

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-317672

(43)Date of publication of application : 16.11.1999

---

(51)Int.Cl. H03M 7/30

G10L 9/18

G11B 20/10

H04H 5/00

---

(21)Application number : 10-308263 (71)Applicant : SAMSUNG  
ELECTRONICS CO LTD

(22)Date of filing : 29.10.1998 (72)Inventor : PARK SUNG-HEE  
KIM YEON-BAE

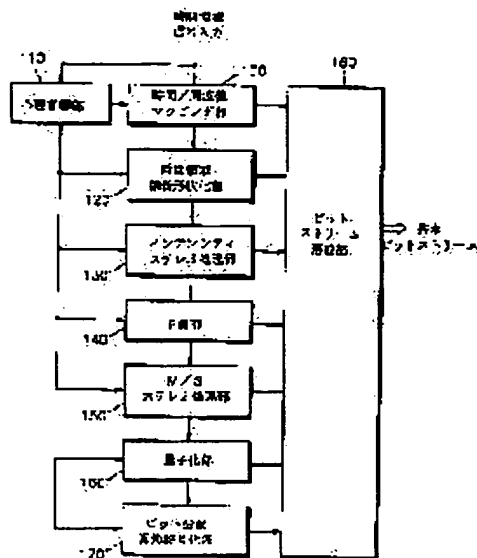
---

(30)Priority

Priority	97 9761605	Priority	20.11.1997	Priority	KR
number :		date :		country :	

---

(54) STEREOPHONIC AUDIO CODING AND DECODING  
METHOD/APPARATUS CAPABLE OF BIT-RATE CONTROL



(57)Abstract:

PROBLEM TO BE SOLVED: To control bit rate by using a bit division arithmetic coding technique, which preferentially codes a more important component of one of two-channel signals and generating a bit stream including a basic hierarchy and many higher hierarchies.

SOLUTION: A bit division arithmetic coding part 170 encodes frequency data value quantized at a quantization part 160 by combining additional information on a relevant band and quantization information given to audio data, based on a bit rate corresponding to each

hierarchy. Meanwhile, a performance similar to AAC is made to be capable of being provided at the highest hierarchy. Then the part 170 limits the band corresponding to a basic hierarchy so as to control the bit rate and also encodes additional information on the basic hierarchy. Then information on the quantized value are encoded in the order covering the highest bit through the lowest one and also successively covering lowest frequency component through the highest one.

일본공개특허공보 평11-317672호(1999.11.16.) 1부.

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平11-317672

(43)公開日 平成11年(1999)11月16日

(51)Int.Cl. <sup>4</sup>	識別記号	F I
H 0 3 M 7/30		H 0 3 M 7/30 Z
G 1 0 L 9/18		G 1 0 L 9/18 M
G 1 1 B 20/10	3 0 1	G 1 1 B 20/10 3 0 1 Z
H 0 4 H 5/00		H 0 4 H 5/00 Z

審査請求 未請求 請求項の数39 O L (全 39 頁)

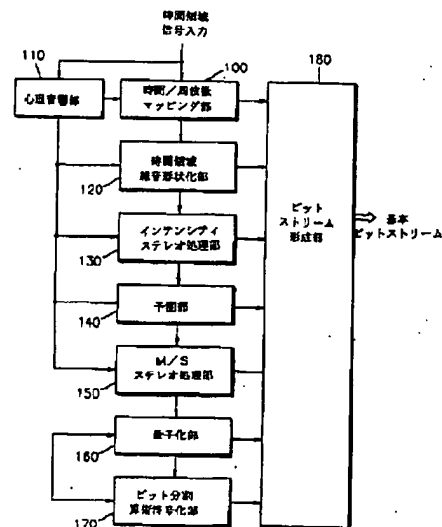
(21)出願番号	特願平10-308263	(71)出願人	390019839 三星電子株式会社 大韓民国京畿道水原市八達区梅灘洞416
(22)出願日	平成10年(1998)10月29日	(72)発明者	朴 成福 大韓民国 ソウル特別市 瑞草区 瑞草1 洞 1642-14番地 韓一アパート MA棟 506號
(31)優先権主張番号	97-61605	(72)発明者	金 延培 大韓民国 京畿道 水原市 權善洞 新東 亜アパート504棟 306號
(32)優先日	1997年11月20日	(74)代理人	弁理士 磯野 道造
(33)優先権主張国	韓国(KR)		

(54)【発明の名称】 ビット率の調節可能なステレオオーディオ符号化/復号化方法及び装置

(57)【要約】

【課題】 ビット率の調節可能なステレオオーディオ符号化/復号化方法及び装置を提供する。

【解決手段】 オーディオ信号を処理して所定の量子化帯域別に量子化する量子化処理、量子化されたデータのうち基本階層に相応するデータを符号化する基本階層符号化処理、符号化した階層の次の高位階層に属するデータ及び符号化できず残置したデータを符号化する階層符号化処理を全階層に対して行い、ビットストリームに形成するビットストリーム形成処理を行い、基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、符号化する階層に相応する付加情報及び量子化したデータを所定のビット数で表現し、重要度が最高の最上位ビットシーケンスから重要度の低い最下位ビットシーケンスの順に算術符号化を行い、左右のチャンネルデータを所定のベクトル単位で交互に符号化する。



【特許請求の範囲】

【請求項1】 オーディオ信号を基本階層及び少なくとも2つの高位階層からなる階層的なビットストリームに符号化するビット率の調節可能なステレオオーディオ信号の符号化方法において、

入力オーディオ信号を信号処理して所定の量子化帯域別に量子化する量子化処理と、

前記量子化処理されたデータのうちの基本階層に相応する量子化したデータを符号化する基本階層符号化処理と、

前記符号化する階層の次の高位階層に属する量子化したデータ及び該符号化する階層に属しながら階層の大きさの制限が原因で符号化できずに残留したデータをそれぞれ符号化する階層符号化処理と、

前記階層符号化処理を全階層に対して行い、ビットストリームを形成するビットストリーム形成処理とを行い、前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理における符号化は、

符号化しようとする階層に相応する付加情報及び量子化したデータを所定数のビットで表現し、重要度が最高のビットからなる最上位ビットシーケンスから重要度の低いビットからなる最下位ビットシーケンスの順に従って所定の確率モデルを用いて算術符号化を行い、ビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に符号化することを特徴とするビット率の調節可能なステレオオーディオ符号化方法。

【請求項2】 前記付加情報は、少なくともスケールファクタ及び算術符号化に使用される確率モデル情報を含むことを特徴とする請求項1記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項3】 前記所定のベクトルは、前記ビット分割された4個の各オーディオチャンネルデータを1つのベクトルにした4次元ベクトルであることを特徴とする請求項1記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項4】 前記4次元ベクトルは、前記ビット分割された当該周波数成分について“0”でない値が符号化されたか否かを示す以前の状態に基づき、2つのサブベクトルに分かれて符号化されることを特徴とする請求項3記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項5】 前記スケールファクタの符号化は、スケールファクタ値のうちの最大値を求める処理と、最初のスケールファクタと前記最大値との差分を求めて算術符号化を行う処理とを行い、

前記最初のスケールファクタ以降のスケールファクタは、算術符号化した直前のスケールファクタとの差分を求

め、該差分を所定の値にマッピングし、該マッピングされた値を算術符号化することを特徴とする請求項2記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項6】 前記マッピングは、

表5. 1を使用することを特徴とする請求項5記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項7】 前記算術符号化は、

確率モデルとして表5. 3及び表5. 4を使用することを特徴とする請求項5記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項8】 前記スケールファクタの符号化は、スケールファクタ値のうちの最大値を求める段階と、各スケールファクタと前記最大値との差分を求めて算術符号化を行う段階とを、行うことを特徴とする請求項2記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項9】 全ての帯域に共通に使用されるヘッダ情報を符号化し、各階層に必要な付加情報と量子化した周波数とをビット分割された情報から構成して階層的に符号化することを特徴とする請求項1記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項10】 前記量子化処理は、時間領域の入力オーディオ信号を周波数領域の信号に切替える処理と、

前記時間/周波数マッピングに基づいて周波数成分に切替えられた入力オーディオ信号を所定のスケールファクタバンド信号で結び付け、前記各スケールファクタバンドにおけるマスキング閾値を計算する処理と、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化処理と、

2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを伝送するインテンシティステレオ処理と、

現フレームの周波数係数の値を予測する予測処理と、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ足した和信号及び引いた差信号に切替えた後、この信号を処理するMiddle/Side ステレオ処理と、

各帯域の量子化雑音がマスキング閾値よりも小さくなるように符号化帯域別に量子化を行う処理とを、行うことを特徴とする請求項1記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項11】 前記量子化したデータは、符号データと大きさデータとからなり、

前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理における符号化は、

前記大きさデータの最上位ビットからなる最上位ビットシーケンスを算術符号化する処理と、

前記符号化した最上位ビットシーケンスのうちの“0”でないデータに該当する符号データを符号化する処理と、

前記符号化していない大きさデータのうちの重要度が最高のビットシーケンスを算術符号化する大きさ符号化処理と、

符号化したビットシーケンスのうちの“0”でない大きさデータに該当する符号データのうちの符号化していない信号データを符号化する信号データ符号化処理と、  
前記大きさ符号化処理及び信号データ符号化処理を前記量子化したデータの各ビットに対して行う処理とを行い、

前記各処理をビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に行うことを特徴とする請求項1記載のビット率の調節可能なステレオオーディオ符号化方法。

【請求項12】 入力オーディオ信号を信号処理して所定の符号化帯域別に量子化する量子化処理部と、

ビット率が調節自在に基本階層に相応する帯域制限を行うと共に付加情報を符号化し、量子化した値に関する情報を最上位ビットから最下位ビットの順に従い且つ低い周波数から高い周波数の順に従って符号化し、左側のチャンネル及び右側のチャンネルを所定のベクトル単位で交互に符号化し、基本階層に対する符号化が終了した後、次の階層に関する付加情報及びオーディオデータの量子化値を符号化して全階層に対してビットストリームが階層的な構造を有するように符号化を行うビット分割算術符号化部と、

前記量子化処理部及びビット分割算術符号化部において作られたデータをまとめてビットストリームを生成するビットストリーム形成部とを、備えたことを特徴とするビット率の調節可能なステレオオーディオ符号化装置。

【請求項13】 前記量子化処理部は、  
時間領域の入力オーディオ信号を周波数領域の信号に切替える時間／周波数マッピング部と、

時間／周波数マッピングに基づいて周波数成分に切替えられた入力オーディオ信号を所定の帯域の信号で結び付け、各信号同士の相互作用によって生じるマスキング現象を用いて各帯域におけるマスキング閾値を計算する心理音響部と、

各帯域の量子化雑音をマスキング閾値と比較しながら量子化を行う量子化部とを、備えたことを特徴とする請求項12記載のビット率の調節可能なステレオオーディオ符号化装置。

【請求項14】 変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化部と、

2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを

伝送するインテンシティステレオ処理部と、

現フレームの周波数係数の値を予測する予測部と、

左側のチャンネル信号及び右側のチャンネル信号をそれぞれ足した和信号及び引いた差信号に切替えた後、この信号を処理するMiddle/Side ステレオ処理部とを、更に具備することを特徴とする請求項13記載のビット率の調節可能なステレオオーディオ符号化装置。

【請求項15】 階層的な構造を有するビット率の調節可能なオーディオデータを復号化する方法において、  
階層的な構造を有するビットストリームにおいて各モジュールに必要なデータを分析する分析処理と、

前記階層的な構造を有するビットストリームが作られた順に従い、少なくともスケールファクタ、算術符号化モデル指数及び量子化したデータを復号化し、このとき、該ビットストリームを構成しているビットの重要度を分析し、重要度の高いビットから重要度の低いビットへ下りながらチャンネル別に交互に該量子化したデータを復号化する復号化処理と、

前記復号化したスケールファクタ及び量子化したデータを用いて元の寸法の信号に復元する信号復元処理と、

前記復元された信号を時間領域の信号に切替える変換処理とを、行うことを特徴とするビット率の調節可能なステレオオーディオ復号化方法。

【請求項16】 前記ビットストリームの符号化過程でMiddle/Side ステレオ処理が施されたか否かを判定した上、該Middle/Side ステレオ処理が施された場合、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号と引いた差信号から元の信号に切替えるMiddle/Side ステレオ処理と、

前記ビットストリームの符号化過程で予測処理が施されたか否かを検査した上、該予測処理が施された場合、現フレームの周波数係数の値を予測する予測処理と、

前記ビットストリームの符号化過程でインテンシティステレオ処理が施されたか否かを判定した上、該インテンシティステレオ処理が施された場合、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルに関する量子化した情報を該一方のチャンネルの値に復元するインテンシティステレオ処理と、

前記ビットストリームの符号化過程で時間領域の雑音形状化処理が施されたか否かを判定した上、該時間領域の雑音形状化処理が施された場合、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化処理とを、更に具備することを特徴とする請求項15記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項17】 前記量子化したデータが符号データと大きさデータとからなるとき、量子化した周波数成分の大きさデータと符号データとを順次復号化し、該大きさデータと符号データとを組合わせて量子化した周波数成

分を復元することを特徴とする請求項15又は16記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項18】 前記復号化処理は、重要度の高い最上位ビットから重要度の低い最下位ビットの順に従って復号化し、

前記信号復元処理は、前記復号化したビット分割されたデータを組合せ、量子化した周波数成分データに復元することを特徴とする請求項15記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項19】 前記復号化処理のデータ復号化は、4次元ベクトル単位で4個のサンプルのビット分割情報を復号化することを特徴とする請求項18記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項20】 前記4次元ベクトル復号化は、ビット分割された当該周波数成分に対し、“0”でない値が符号化されたか否かを示す以前の状態に基づいて符号化した2つのサブベクトルを算術復号化すると共に、各サンプルの符号化状態に基づいて復号化した2つのサブベクトルを4次元ベクトルに復元することを特徴とする請求項19記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項21】 各量子化した周波数成分のビット分割されたデータを最上位ビットから復号化するとき、分割されたビット値が“0”であればそのまま通過してから最初に“1”が出たときに符号データを算術復号化することを特徴とする請求項17記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項22】 前記スケールファクタ復号化は、ビットストリームからまずスケールファクタに対する最大値を復号化し、各量子化帯域のスケールファクタの差分信号を算術復号化した後に、最大値から差分信号を引いてスケールファクタを復号化することを特徴とする請求項15記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項23】 前記スケールファクタ復号化は、ビットストリームからスケールファクタに対する最大値を復号化する処理と、前記最大値と復号化しようとするスケールファクタとの差分をマッピングした値を算術復号化し、前記マッピングされた値から前記差分を逆マッピングして前記差分を得る処理と、前記差分を最大値から引いて最初のスケールファクタを求めると共に、残りの帯域のスケールファクタは以前のスケールファクタから差分を引いてスケールファクタを求める処理とを、行うことを特徴とする請求項15記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項24】 前記算術符号化モデル指数復号化は、ビットストリームからまず算術符号化モデル指数に対す

る最小値を復号化し、各階層の付加情報から各算術符号化モデル指数の差分信号を復号化した後に、最小値と差分信号とを和して算術符号化モデル指数を復号化することを特徴とする請求項15記載のビット率の調節可能なステレオオーディオ復号化方法。

【請求項25】 階層的な構造を有するビット率の調節可能なオーディオデータを復号化する装置において、階層的な構造を有するビットストリームで各モジュールに必要なデータを分析するビットストリーム分析部と、前記階層的な構造を有するビットストリームが作られた順に従って少なくともスケールファクタ、算術符号化モデル指数及び量子化したデータを復号化し、このとき、前記ビットストリームを構成しているビットの重要度を分析した上で、重要度の高いビットから重要度の低いビットへ下りながらチャンネル別に交互に前記量子化したデータを復号化する復号化部と、

前記復号化したスケールファクタ及び量子化したデータを用いて元の寸法の信号に復元する信号復元部と、前記復元された信号を時間領域の信号に切替える周波数／時間マッピング部とを、備えたことを特徴とするビット率の調節可能なステレオオーディオ復号化装置。

【請求項26】 ビットストリームの符号化過程でMiddle/Side ステレオ処理が施されたか否かを検査した上、該Middle/Side ステレオ処理が施された場合左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号と引いた差信号から元の信号に切替えるMiddle/Side ステレオ処理部と、

ビットストリームの符号化過程で予測処理が施されたか否かを検査した上、予測処理が施された場合現フレームの周波数係数の値を予測する予測部と、ビットストリームの符号化過程でインテンシティステレオ処理が施されたか否かを検査した上、インテンシティステレオ処理が施された場合に2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルに関する量子化した情報を前記一方のチャンネル値に復元するインテンシティステレオ処理部と、前記ビットストリームの符号化過程で時間領域雑音形状化処理が施されたか否かを判定した上、時間領域雑音形状化処理が施された場合、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化部とを、更に具備することを特徴とする請求項25記載のビット率の調節可能なステレオオーディオ復号化装置。

【請求項27】 オーディオ信号を基本階層及び少なくとも2つの高位階層からなる階層的なビットストリームに符号化するビット率の調節可能なオーディオ信号符号化方法において、

入力オーディオ信号を信号処理して所定の量子化帯域別に量子化する量子化処理と、

前記量子化処理されたデータのうちの基本階層に相応する量子化したデータを符号化する基本階層符号化処理と、

前記符号化した階層の次の高位階層に属する量子化したデータ及び前記符号化した階層に属しながら階層の大きさの制限が原因で符号化できずに残留したデータをそれぞれ符号化する階層符号化処理と、

前記階層符号化処理を全階層に対して行い、ビットストリームに形成するビットストリーム形成処理とを行い、前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理における符号化は、

符号化しようとする階層に相応する付加情報及び量子化したデータを所定数のビットで表現し、重要度が最高のビットからなる最上位ビットシーケンスから重要度の低いビットからなる最下位ビットシーケンスの順に従って所定の確率モデルを用いて算術符号化を行い、ビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に符号化することの特徴とするビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項28】 前記付加情報は、

少なくともスケールファクタ及び算術符号化に使用される確率モデル情報を含み、

前記スケールファクタの符号化は、

スケールファクタ値のうちの最大値を求める処理と、

最初のスケールファクタと前記最大値との差分を求めて算術符号化する処理と、

前記最初のスケールファクタ以降のスケールファクタは算術符号化した直前のスケールファクタとの差分を求め、前記差分を所定の値でマッピングして該マッピングされた値を算術符号化する処理とを、行うことを特徴とするビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項29】 前記所定のベクトルは、

前記ビット分割された4個の各オーディオチャンネルデータを1つのベクトルにした4次元ベクトルであり、

前記4次元ベクトルは、

ビット分割された該当周波数成分に対し、“0”でない値が符号化されたか否かを示す以前の状態に基づき、2つのサブベクトルに分かれて符号化されることを特徴とする請求項27記載のビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項30】 前記量子化処理は、

時間領域の入力オーディオ信号を周波数領域の信号に切替える処理と、

前記時間／周波数マッピングに基づいて周波数成分に切替えられた入力オーディオ信号を所定のスケールファク

タバンド信号で結び付け、前記各スケールファクタバンドにおけるマスキング閾値を計算する処理と、

変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化処理と、

2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを伝送するインテンシティステレオ処理と、

現フレームの周波数係数の値を予測する予測処理と、

左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号及び引いた差信号に切替えた後に、この信号を処理するMiddle/Side ステレオ処理と、

各帯域の量子化雑音がマスキング閾値よりも小さくなるように所定の符号化帯域別に量子化を行う処理とを、行うことを特徴とする請求項27記載のビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項31】 前記量子化したデータは、符号データと大きさデータとからなり、

前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、

前記大きさデータの最上位ビットからなる最上位ビットシーケンスを算術符号化する処理と、

前記符号化した最上位ビットシーケンスのうちの“0”でないデータに該当する符号データを符号化する処理と、

前記符号化していない大きさデータのうちの重要度が最高のビットシーケンスを算術符号化する大きさ符号化処理と、

符号化したビットシーケンスのうちの“0”でない大きさデータに該当する信号データのうちの符号化していない信号データを符号化する信号データ符号化処理と、前記大きさ符号化処理及び信号データ符号化処理を前記量子化したデータの各ビットに対して行う処理とを含み、該各処理をビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に行うことを特徴とする請求項27記載のビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項32】 階層的な構造を有するビット率の調節可能なオーディオデータを復号化する方法において、

階層的な構造を有するビットストリームにおいて各モジュールに必要なデータを分析する分析処理と、

前記階層的な構造を有するビットストリームが作られた順に従って少なくともスケールファクタ、算術符号化モデル指数及び量子化したデータを復号化し、このとき、前記ビットストリームを構成しているビットの重要度を

分析した上、重要度の高いビットから重要度の低いビットへ下りながらチャンネル別に交互に前記量子化したデータを復号化する復号化処理と、

前記復号化したスケールファクタ及び量子化したデータを用いて元の寸法の信号に復元する信号復元処理と、前記復元された信号を時間領域の信号に切替える変換処理とを、含むことを特徴とするビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項33】 前記ビットストリームの符号化過程でMiddle/Side ステレオ処理が施されたか否かを判定した上、Middle/Side ステレオ処理が施された場合、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号及び引いた差信号から元の信号に切替えるMiddle/Side ステレオ処理と、

前記ビットストリームの符号化過程で予測処理が施されたか否かを検査した上、予測処理が施された場合、現フレームの周波数係数の値を予測する予測処理と、前記ビットストリームの符号化過程でインテンシティステレオ処理が施されたか否かを検査した上、インテンシティステレオ処理が施された場合、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルに関する量子化した情報を前記一方のチャンネルの値に復元するインテンシティステレオ処理と、

前記ビットストリームの符号化過程で時間領域雑音形状化処理が施されたか否かを判定した上、時間領域雑音形状化処理が施された場合、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化処理とを、更に具備することを特徴とする請求項32記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項34】 前記量子化したデータが符号データと大きさデータとからなるとき、

量子化した周波数成分の大きさ値及び符号ビットの順に従って復号化を行い、大きさ及び符号ビットを組合わせて量子化した周波数成分を復元することを特徴とする請求項32又は33記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項35】 前記復号化処理は、重要度の高い最上位ビットから重要度の低い最下位ビットの順に従って復号化を行い、

前記信号復元処理は、前記復号化したビット分割されたデータを組合わせ量子化した周波数成分データに復元することを特徴とする請求項32記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項36】 復号化処理のデータ復号化は、4次元ベクトル単位で4個のサンプルのビット分割情報を復号化し、

前記4次元ベクトル復号化は、

ビット分割された当該周波数成分に対し、“0”でない値が符号化されたか否かを示す以前の状態に基づいて符号化した2つのサブベクトルを算術復号化し、各サンプルの符号化の状態に基づいて復号化した2つのサブベクトルを4次元ベクトルに復元することを特徴とする請求項35記載のステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項37】 各量子化した周波数成分のビット分割されたデータを最上位ビットから復号化していく最中に、分割されたビット値が“0”であればそのまま通過してから最初に“1”が出たときに符号値を算術復号化することを特徴とする請求項34記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項38】 前記スケールファクタ復号化は、ビットストリームからまずスケールファクタに対する最大値を復号化し、各量子化帯域のスケールファクタの差分信号を算術復号化した後に、最大値から差分信号を引いてスケールファクタを復号化することを特徴とする請求項32記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【請求項39】 前記スケールファクタ復号化は、ビットストリームからスケールファクタに対する最大値を復号化する処理と、

前記最大値と復号化しようとするスケールファクタとの差分をマッピングした値を算術復号化し、前記マッピングされた値から前記差分を逆マッピングして前記差分を得る処理と、

前記差分を最大値から引いて最初のスケールファクタを求め、残りの帯域のスケールファクタは、以前のスケールファクタから差分を引いてスケールファクタを求める処理とを、行うことを特徴とする請求項32記載のビット率の調節可能なステレオオーディオ復号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体、

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、オーディオ信号の符号化及び復号化に用いられ、特にビット率の調節可能なステレオオーディオ符号化／復号化方法及び装置に関するものである、

【0002】

【従来の技術】従来、このような分野の技術としては、



例えば、次のような文献に記載されるものがあった。  
文献：K.Brandenburg,et.al.,“First Ideas on Scalable Audio Coding”, 97thAES-Convention,preprint 3924,San Francisco,1994. K.Brandenburg,et al.,“A Two-or Three-Stage Bit Rate Scalable Audio Coding System”,99th AES-Convention,preprint 4132,NewYork,1995  
前記文献には、従来のビット率の調節可能なオーディオ符号化／復号化装置は、主に1チャンネルのモノラル信号に対するスケラビリティのみを考慮してきたことが示されている。ところが、現在、符号化効率のみ考慮して作られたMP E G／オーディオ標準(MPEG Committee ISO /IEC/JTC1/SC29/WG11,Information technology-Coding of moving pictures and associated audio for data s torage media to about 1.5Mbit/s-Part3:Audio,ISO/IEC IS 11172-3,1993)や、A C - 2 / A C - 3方式(Dolby,“Dolby AC-3 Multi-Channel Audio Coding-Submission to the Grand Alliance Audio Specialist Group”,Dolby Lab, Aug.1993)では、モノチャンネルのみならず、ステレオやそれ以上の多チャンネル信号まで処理している。実際、多くの音楽信号は、ステレオ信号から構成されている。従って、インターネットや通信網を通じたサービスにおいて、2チャンネル以上のビットストリームから構成された信号に適用し得るスケラブルオーディオコーデックが望まれている。

【0003】一般に、音楽信号は、ステレオ信号の形式になっている。このステレオ信号は、コンパクトディスク(CD)、通信網、又は放送網を介して提供されており、今後、マルチメディア環境からも提供されることが予想される。しかし、従来のスケラブルオーディオコーデックは、主にモノラル信号のみに対して考慮されているため、ステレオ信号は未だ処理し得ない、散えてステレオ信号を処理しようとするれば、まず一方のチャンネルに対する信号を送り切ったうえで、他方のチャンネルに対する信号を送る方式しかない。この場合、2チャンネルのビット発生量が常に等しいわけではないので、ステレオ信号に対してビット率が低下すると共に、ビット率の調節可能なオーディオコーデックの性能が著しく劣化する。

【0004】

【発明が解決しようとする課題】本発明が果たそうとする技術的課題は、ビットストリームを構成するに当り、2つのチャンネル信号のうちの重要度が最高の信号成分を優先的に符号化するビット分割算術符号化(BSAC, Bit-Sliced Arithmetic Coding)技法を利用し、基本階層(Base Layer)を基に多数の高位階層(Enhancement Layer)から構成されたビットストリームを生成することにより、ビット率の調節可能なステレオデジタルオーディオデータ符号化／復号化方法及び装置を提供することである。

【0005】

【課題を解決するための手段】前記技術的課題を解決す

るための本発明に係るオーディオ符号化方法は、オーディオ信号を、基本階層及び少なくとも2つの高位階層からなる階層的なビットストリームに符号化するビット率の調節可能なステレオオーディオ信号の符号化方法において、入力オーディオ信号を信号処理して所定の量子化帯域別に量子化する量子化処理と、前記量子化処理されたデータのうちの基本階層に相応する量子化したデータを符号化する基本階層符号化処理と、前記符号化した階層の次の高位階層に属する量子化したデータ及び前記符号化した階層に属しながら、階層の大きさの制限が原因で符号化できずに残留した量子化したデータを符号化する階層符号化処理と、前記階層符号化処理を全階層に対して行い、ビットストリームに形成するビットストリーム形成処理とを含むようにしている。前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、符号化しようとする階層に相応する付加情報及び量子化したデータを所定数のビットで表現し、重要度が最高のビットからなる最上位ビットシーケンスから重要度の低い最下位ビットからなるビットシーケンスの順に従って所定の確率モデルを用い算術符号化を行うものの、但し、ビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に符号化する。

【0006】前記付加情報は、少なくともスケールファクタ及び算術符号化に使用される確率モデル情報を含む。前記所定のベクトルは、前記ビット分割された4個の各オーディオチャンネルデータを1つのベクトルにした4次元ベクトルであることが好ましく、この4次元ベクトルは、ビット分割された当該周波数成分について“0”でない値が符号化されたか否かを示す以前の状態に基づき、2つのサブベクトルに分かれて符号化されるようになっている。前記スケールファクタの符号化は、スケールファクタ値のうちの最大値を求める処理と、最初のスケールファクタと前記最大値との差分を求め、算術符号化を行う処理と、前記最初のスケールファクタ以降のスケールファクタは、算術符号化した直前のスケールファクタとの差分を求め、前記差分を所定の値にマッピングし、前記マッピングされた値を算術符号化する処理とを、行うようにしている。

【0007】前記スケールファクタの符号化は、スケールファクタ値のうちの最大値を求める処理と、各スケールファクタと前記最大値に対する差分を求め、算術符号化を行う処理とを、行うようにしている。前記符号化方法は、全ての帯域に共通に使用されるヘッダ情報を符号化し、各階層に必要な付加情報と量子化した周波数とをビット分割された情報から構成して階層的に符号化するようにしている。前記量子化処理は、時間領域の入力オーディオ信号を周波数領域の信号に切替える処理と、前記時間／周波数マッピングに基づいて周波数成分に切替えられた入力オーディオ信号を所定のスケールファクタ

バンド信号で結び付け、前記各スケールファクタバンドにおけるマスキング閾値を計算する処理と、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化(TNS: Temporal Noise Shaping)処理と、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを送送するインテンシティステレオ処理と、現フレームの周波数係数の値を予測する予測処理と、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号及び引いた差信号に切替えた後に、この信号を処理するM/Sステレオ処理と、各帯域の量子化雑音がマスキング閾値よりも小さいように所定の符号化帯域別に量子化を行う処理とを、行うようにしている。

【0008】前記量子化したデータは、信号データと大ききデータとからなり、前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、前記大ききデータの最上位ビットからなる最上位ビットシーケンスを算術符号化する処理と、前記符号化した最上位ビットシーケンスのうちの“0”でないデータに該当する信号データを符号化する処理と、前記符号化していない大ききデータのうちの重要度が最高のビットシーケンスを算術符号化する大きき符号化処理と、符号化したビットシーケンスのうちの“0”でない大ききデータに該当する信号データのうちの符号化していない信号データを符号化する信号データ符号化処理と、前記大きき符号化処理及び信号データ符号化処理を前記量子化したデータの各ビットに対して行う処理とを含み、前記各処理をビット分割された左側のチャンネルデータ及び右側のチャンネルデータを所定のベクトル単位で交互に行う。

【0009】前記技術的課題を解決するための本発明に係るオーディオ符号化装置は、入力オーディオ信号を信号処理して所定の符号化帯域別に量子化する量子化処理部と、ビット率が調節自在に基本階層に相応する帯域制限をすると共に、付加情報を符号化し、量子化した値に関する情報を最上位ビットから、又、低い周波数から高い周波数の順に従って符号化し、左側のチャンネル及び右側のチャンネルを所定のベクトル単位で交互に符号化し、基本階層に対する符号化が終了すると、次の階層に関する付加情報及びオーディオデータの量子化値を符号化し、全階層に対しビットストリームが階層的な構造を有するように符号化を行うビット分割算術符号化部と、前記量子化部及びビット分割算術符号化部において作られたデータをまとめてビットストリームを生成するビットストリーム形成部とを含むようにしている。

【0010】前記量子化処理部は、時間領域の入力オーディオ信号を周波数領域の信号に切替える時間/周波数マッピング部と、時間/周波数マッピングに基づいて周波数成分に切替えられた入力オーディオ信号を所定の帯

域の信号で結び付け、各信号同士の相互作用によって生じるマスキング現象を用いて各帯域におけるマスキング閾値を計算する心理音響部と、各帯域の量子化雑音をマスキング閾値と比較しつつ、量子化を行う量子化部とを含むようにしている。そして、前記符号化装置は、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化部と、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを送送するインテンシティステレオ処理部と、現フレームの周波数係数の値を予測する予測部と、左側のチャンネル信号及び右側のチャンネル信号をそれぞれ和した和信号及び引いた差信号に切替えた後に、この信号を処理するMiddle/Side (以下、「M/S」という)ステレオ処理部とを、更に具備するようにしている。

【0011】前記技術的課題を解決するための本発明に係るオーディオ復号化方法は、階層的な構造を有するビット率の調節可能なオーディオデータを復号化する方法において、階層的な構造を有するビットストリームにおいて各モジュールに必要なデータを分析する分析処理と、前記階層的な構造を有するビットストリームが作られた順に従い、少なくともスケールファクタ、算術符号化モデル指数及び量子化したデータを復号化するものの、但し、前記ビットストリームを成しているビットの重要度を分析した上、重要度の高いビットから重要度の低いビットへ下りながらチャンネル別に交互に前記量子化したデータを復号化する復号化処理と、前記復号化したスケールファクタ及び量子化したデータを用いて元の寸法の信号に復元する信号復元処理と、前記復元された信号を時間領域の信号に切替える変換処理とを含むようにしている。

【0012】前記復号化方法は、前記ビットストリームの符号化過程でM/Sステレオ処理が施されたか否かを判定した上、M/Sステレオ処理が施された場合、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和した和信号と引いた差信号から元の信号に切替えるM/Sステレオ処理と、前記ビットストリームの符号化過程で予測処理が施されたか否かを検査した上、予測処理が施された場合、現フレームの周波数係数の値を予測する予測処理と、前記ビットストリームの符号化過程でインテンシティステレオ処理が施されたか否かを判定した上、インテンシティステレオ処理が施された場合、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルに関する量子化した情報を前記一方のチャンネルの値に復元するインテンシティステレオ処理と、前記ビットストリームの符号化過程で時間領域の雑音形状化処理が施されたか否かを判定した上、時間領域の雑音形状化処理が施された場合、変換の各ウィ

ドフ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化処理とを、更に具備するようにしている。

【0013】前記復号化方法は、前記量子化したデータが符号データと大きさデータとからなるとき、量子化した周波数成分の大きさ値と符号ビットとを順次復号化し、大きさ及び符号ビットを組合わせて量子化した周波数成分を復元するようにしている。前記復号化処理は、重要度の高い最上位ビットから重要度の低い最下位ビットの順に従って復号化し、前記信号復元処理は、前記復号化したビット分割されたデータを組合せ、量子化した周波数成分データに復元するようにしている。復号化処理のデータ復号化は、4次元ベクトル単位で4個のサンプルのビット分割情報を復号化するようにし、前記4次元ベクトル復号化は、ビット分割された当該周波数成分に対し、“0”でない値が符号化されたか否かを示す以前の状態に基づいて符号化した2つのサブベクトルを算術復号化すると共に、各サンプルの符号化状態に基づいて復号化した2つのサブベクトルを4次元ベクトルに復元するようにしている。

【0014】更に、前記復号化方法は、各量子化した周波数成分のビット分割されたデータを最上位ビットから復号化していく最中に、分割されたビット値が“0”であれば、そのまま通過してから最初に“1”が出たときに符号値を算術復号化するようにしている。前記スケールファクタ復号化は、ビットストリームからまずスケールファクタに対する最大値を復号化し、各量子化帯域のスケールファクタの差分信号を算術復号化した後に、最大値から差分信号を引いてスケールファクタを復号化するようにしている。更に、前記スケールファクタ復号化は、ビットストリームからスケールファクタに対する最大値を復号化する処理と、前記最大値と復号化しようとするスケールファクタとの差分をマッピングした値を算術復号化し、前記マッピングされた値から前記差分を逆マッピングして前記差分を得る処理と、前記差分を最大値から引いて最初のスケールファクタを求めると共に、残りの帯域のスケールファクタは以前のスケールファクタから差分を引いてスケールファクタを求める処理とを、行うようにしている。前記算術符号化モデル指数復号化は、ビットストリームからまず算術符号化モデル指数に対する最小値を復号化し、各階層の付加情報から各算術符号化モデル指数の差分信号を復号化した後に、最小値と差分信号とを和して算術符号化モデル指数を復号化するようにしている。

【0015】前記技術的課題を解決するための本発明に係るオーディオ復号化装置は、階層的な構造を有するビット率の調節可能なオーディオデータを復号化する装置において、階層的な構造を有するビットストリームで各モジュールに必要なデータを分析するビットストリーム分析部と、前記階層的な構造を有するビットストリーム

が作られた順に従い、少なくともスケールファクタ、算術符号化モデル指数及び量子化したデータを復号化するものの、但し、前記ビットストリームを成しているビットの重要度を分析した上で、重要度の高いビットから重要度の低いビットへ下りながらチャンネル別に交互に前記量子化したデータを復号化する復号化部と、前記復号化したスケールファクタ及び量子化したデータを用いて元の寸法の信号に復元する信号復元部と、前記復元された信号を時間領域の信号に切替える周波数/時間マッピング部とを含むようにしている。

【0016】前記復号化装置は、ビットストリームの符号化過程でM/Sステレオ処理が施されたか否かを判定した上、M/Sステレオ処理が施された場合、左側のチャンネルの信号及び右側のチャンネルの信号をそれぞれ和信号と引いた差分信号から元の信号に切替えるM/Sステレオ処理部と、ビットストリームの符号化過程で予測処理が施されたか否かを判定した上、予測処理が施された場合、現フレームの周波数係数の値を予測する予測部と、ビットストリームの符号化過程でインテンシティステレオ処理が施されたか否かを判定した上、インテンシティステレオ処理が施された場合、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルに関する量子化した情報を前記一方のチャンネル値に復元するインテンシティステレオ処理部と、前記ビットストリームの符号化過程で時間領域雑音形状化処理が施されたか否かを判定した上、時間領域雑音形状化処理が施された場合、変換の各ウィンドウ内において量子化雑音の時間的な形状を制御するために使用される時間領域雑音形状化部とを、更に具備するようにしている。

【0017】そして、本発明は、コンピュータで実行し得るプログラムによって作成可能である。更に、コンピュータで使用される媒体から前記プログラムを動作させる汎用のデジタルコンピュータで具現できる。前記媒体は、マグネチック記憶媒体（例えば、ROM、フロッピーディスク、ハードディスク等）、光学的記憶媒体（例えば、CD-ROM、DVD等）及びキャリアウェーブ（例えば、インターネットを介した伝送）等の記憶媒体を含む。

【0018】例えば、本発明は、オーディオ信号を基本階層及び予め決定された数の高位階層からなる階層的なビットストリームに符号化するビット率の調節可能なオーディオ信号符号化方法において、入力オーディオ信号を信号処理して所定の量子化帯域別に量子化する量子化処理と、前記量子化処理されたデータのうちの基本階層に相応する量子化したデータを符号化する基本階層符号化処理と、前記符号化した階層の次の高位階層に属する量子化したデータ及び前記符号化した階層に属しながら階層の大きさの制限を理由に符号化できずに残っている

量子化したデータを符号化する階層符号化処理と、前記階層符号化処理を全階層に対して行い、ビットストリームに形成するビットストリーム形成処理とを含むようにしている。そして、前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、符号化しようとする階層に相応する付加情報及び量子化したデータを所定数のビットで表現し、重要度が最高のビットからなる最上位ビットシーケンスから重要度の低い最下位ビットからなるビットシーケンスの順に従って所定の確率モデルに基づいて算術符号化するものの、但し、ビット分割された左側のチャネルデータ及び右側のチャネルデータが、所定のベクトル単位で交互に符号化するビット率の調節可能なステレオオーディオ符号化方法を実行するためのプログラムを記録したコンピュータで読取できる記録媒体に記憶されることがある。

【0019】

【発明の実施の形態】以下、添付の図面に基づいて本発明の実施形態を更に詳細に説明する。本実施形態は、ビット分割算術符号化(BSAC: Bit-Sliced Arithmetic Coding)技法を利用し、ビット率の調節可能なステレオデジタルオーディオデータを符号化及び復号化するものである。即ち、本実施形態は、既存のオーディオ符号化/復号化に使用される基本モジュールをそのまま使用しつつ、単に無損失符号化モジュールのみを符号化方式に代替する。本実施形態は、このように構成されたビット率の調節可能な符号化/復号化器をステレオ信号にも適用し得るように拡張したものである。

【0020】図1は、本発明の実施形態を示す符号化装置の構成ブロック図である。この符号化装置は、時間/周波数マッピング部100、心理音響部110、時間領域雑音形状化部120、インテンシティステレオ処理部130、予測部140、M/Sステレオ処理部150、量子化部160、ビット分割算術符号化部170及びビットストリーム形成部180から構成されている。

【0021】人間の聴覚特性のうち、デジタルオーディオ信号の符号化における最も重要な現象として使用されるものは、マスキング(masking)効果及び臨界帯域(critical band)の性質である。マスキング効果とは、或る信号(例えば、音)によって他の信号(例えば、音)が聞こえなくなる現象である。例えば、駅を電車が通過するとき、低い声で話しをすると、それが電車の音にぎ込まれて聞こえなくなる現象である。臨界帯域とは、人間の可聴周波数範囲内において、或る帯域別に信号を認知する現象を言い、様々な特性を有している。この特性の1つが、同じ大きさの雑音に対し、或る臨界帯域内にあるときと臨界帯域を外れたときとで、人間が認知する雑音の大きさが違うということである。この場合、臨界帯域を外れたときの雑音に対する認知度の方が高い。聴覚特性を用いて符号化を行うことは、基本的に前記2種類の特性を利用して1つの臨界帯域内において雑音がどれ

だけ割当てられるかを計算した後、量子化時に量子化雑音をそれに見合う分だけ生じさせることにより、符号化による情報の損失を最小化することを意味する。

【0022】時間/周波数マッピング部100は、時間領域の入力オーディオ信号を周波数領域の信号に切替えるものである。心理音響部110は、時間/周波数マッピング部100に基づいて時間領域から周波数領域に切替えられたオーディオ信号を適切な帯域(scale factor band)の信号で結び付け、各信号の相互作用によって生じるマスキング現象を用いて各帯域におけるマスキング閾値(masking threshold)を計算するものである。時間領域雑音形状化(TNS: Temporal Noise Shaping)部120は、変換の各ウィンドウ内で量子化雑音の時間的形狀を制御するために使用されるものであり、周波数データのフィルタリング過程を適用することにより、時間領域雑音形状化が可能となる。前記モジュールは、符号化器において選択的に使用可能なモジュールである。インテンシティステレオ処理部130は、ステレオ信号を一層効率良く処理するためのモジュールであり、2つのチャネルのうちのいずれか一方のチャネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャネルはスケールファクタのみを伝送するものである。前記モジュールは符号化器において必ずしも使用すべきモジュールではなく、種々の事項を考慮した上、各スケールファクタバンド単位に使用可否を判断することができる。

【0023】予測部140は、現フレームの周波数係数の値を予測するものである。このように予測された値と実際の周波数成分との差分を量子化して符号化することにより、使用されるビット発生量を減らすことができる。しかし、予測部140は、フレーム単位で選択的に使用することができる。即ち、予測部140を使用すれば、次の周波数係数を予測するときに複雑度が高まるので、使用しないこともある。又、場合によっては、予測による差分が元の信号よりも大きくなる確率を有しているため、予測することにより、実際に生じたビット発生量が予測しない時よりも大きくなることがあるが、このときは予測部140を使用しない。M/Sステレオ処理部150は、ステレオ信号を一層効率良く処理するためのものであり、左側のチャネル信号と右側のチャネル信号とをそれぞれ和信号と引いた差信号とに切替えた後、この信号を処理する。前記モジュールも、符号化器において必ずしも使用すべきものではなく、符号化器において色々な事項を考慮した上で、各スケールファクタバンド単位で使用可否を判断することができる。量子化部160は、人間が聞いたことを感じさせないために、各帯域の量子化雑音の大きさがマスキング閾値よりも小さくなるように各帯域の周波数信号をスカラー量子化するものであり、音響心理部110で計算したマスキング閾値と各帯域で生じさせる雑音との割合であるNM

R(Noise-to-Mask Ratio)を用い、全帯域のNMR値が0dB以下になるように量子化を行うものである。NMR値が0dB以下であるということは、量子化雑音に比べてマスキング値が高いことを示すが、これは量子化雑音が人に聞こえないという意味である。

【0024】ビット分割算術符号化部170は、本実施形態の核心モジュールであり、MPEG-2AACのような既存のオーディオコーデックは、スケーラビリティを提供し得ないので、AACの無損失符号化部に対する代案として使用することができる。スケーラブルオーディオコーデックが具現できるように、量子化部160で量子化した周波数データ値を各階層に該当するビット率に基づいて該当帯域の付加情報とオーディオデータに対する量子化情報とを組合せて符号化する。又、このスケーラビリティの機能以外に、最高位階層ではAACに類似の性能を提供できるようにする。ビット分割算術符号化部170の機能について、更に詳細に説明すれば、ビット率が調節可能になるように基本階層に相応する帯域制限をすると共に、この基本階層に関する付加情報(side information)を符号化する。そして、量子化した値に関する情報を最上位ビットから最下位ビットの順に、且つ低い周波数成分から高い周波数成分の順に従って符号化する。又、左側のチャンネルと右側のチャンネルとを所定のベクトル単位で交互に符号化し、基本階層に対する符号化を行う。前記基本階層に対する符号化が完了すると、次の階層に関する付加情報とオーディオデータの量子化値とを符号化し、ビットストリームが階層的構造を有するようにする。ビットストリーム形成部180は、符号化器の各モジュールで作られた情報を集め、スケーラブルコーデックに適するように作られた所定のシンタキスト(syntax)に基づいてビットストリームを構成するものである。

【0025】図2は、ビットストリーム形成部180で作られたビットストリームの構造を示すものである。この図に示すように、ビットストリームはビット率によって下位階層のビットストリームが高位階層のビットストリームに含まれている階層構造になっている。従来では、まず付加情報を符号化した後に残りの情報を符号化してビットストリームを作っていたが、本実施形態では、図2に示すように、各階層に必要な付加情報は各階層別に分けられ、階層別に符号化される。又、既存の符号化方式では、量子化したデータを全てサンプル単位に順次符号化する。しかし、本実施形態では、量子化したデータを2進データで表現し、2進データの最上位ビット(MSB)から符号化し、許容可能なビット量内でビットストリームを構成する。

【0026】図3は、本発明の実施形態を示す復号化装置の構成ブロック図である。この復号化装置は、ビットストリーム分析部300、ビット分割算術復号化部310、逆量子化部320、M/Sステレオ処理部330、

予測部340、インテンシティストレオ処理部350、時間領域雑音形状化部360及び周波数/時間マッピング部370から構成されている。

【0027】ビットストリーム分析部300は、入力されたビットストリームでビットストリームが作られた順に従ってヘッダ情報と符号化したデータとを分離し、各モジュールへ送るものである。ビット分割算術復号化部310は、入力されたビットストリームでビットストリームが作られた順に従って付加情報とビット分割された量子化したデータ等を復号化し、逆量子化器320へ送るものである。M/Sステレオ処理部330は、ステレオ信号に限って適用されるモジュールであり、符号化器でM/Sステレオ処理を行ったスケールファクタバンドに対して相応する処理を施すものである。予測部340は、符号化器で予測が行われた場合、以前のフレームで復号化したデータを用いて符号化器に等しい過程の予測を通じて同一値を推すものであり、この予測された値とビットストリーム分析部で復号化した差分信号との和を求めることにより、元の周波数成分を復元するものである。

【0028】インテンシティストレオ処理部350は、ステレオ信号に限って適用されるモジュールであり、符号化器でインテンシティストレオ処理を行ったスケールファクタバンドに対して相応する処理を施すものである。時間領域雑音形状化部360は、変換の各ウィンドウ内で量子化雑音の時間的な形状を制御するために使用されるものであり、相応する処理を施すものである。復号化したデータに対する以降の処理過程は、AAC標準案等の既存のオーディオアルゴリズムのような処理モジュールを経て時間領域の信号に復元することになる。まず、逆量子化部320では、復号化したスケールファクタと量子化したデータとを元の寸法の信号に復元する。そして、周波数/時間マッピング部370は、周波数領域のオーディオ信号を再び時間領域の信号に切替え、使用者が再生できるようにするものである。

【0029】次に、本実施形態のオーディオ符号化動作について説明する。入力されたオーディオ信号は、時間/周波数マッピング部100でMDCT(Modified Discrete Cosine Transform)によって周波数領域の信号に切替えられる。そして、心理音響部110は、周波数信号を適切なスケールファクタバンドで結び付け、マスキング閾値を得る。又、前記周波数領域の信号に切替えられたオーディオ信号は、符号化効率を高めるためのモジュール、即ち、TNS部120、インテンシティストレオ処理部130、予測部140、M/Sステレオ処理部150を経つつ、一層効率良く圧縮された信号になる。量子化部160は、前記モジュールを介して処理された信号を許容可能なビット数内で、人間が聞いたことを感じさせないように、各スケールファクタの量子化雑音の大きさをマスキング閾値と比較していきながら、マスキ

ング閾値よりも小さくなるようにスカラー量子化する。  
前記量子化過程を経てからは、各スケールファクタバンドに対するスケールファクタと量子化した周波数値とが生成される。

【0030】一般に、人間の心理音響学的な側面で、低い周波数は精密な間隔の周波数成分を容易に区別するが、周波数が高くなるにつれて区分できる周波数間隔が広くなる。そこで、スケールファクタバンドの帯域幅は、周波数帯域が高まるに従って次第に大きくなる。しかし、符号化を行うときは、符号化過程の容易性のために、均一でない帯域幅をもったスケールファクタバンドを使用せず、大きさが一定の符号化帯域に変えて使用する。符号化帯域は32個の量子化した周波数係数値を含む。AACのような符号化効率のみを考慮した既存の符号化器／復号化器は、ステレオ信号を処理するとき、まず左側のチャンネルと右側のチャンネルとに共通して使用される情報をヘッダ位置で符号化する。その後、左側のチャンネルのデータを符号化した後、右側のチャンネルのデータを符号化する。即ち、ヘッダ、左側のチャンネル、右側のチャンネルの順に従って符号化が進む。

【0031】このように、ヘッダを処理して左側のチャンネルと右側のチャンネルとに対する情報が重要度によらずに送られる時、ビット率が低くなると、後方向に位置した右側のチャンネルに対する信号から無くなるため、人間

が感じ取れる性能の低下が激しい。しかし、本実施形態に係るステレオオーディオ符号化装置は、チャンネル別に付加情報を符号化する。即ち、前記チャンネル別付加情報は、ビット分割算術符号化部170を介して左側のチャンネルと右側のチャンネルの順に従って交互に符号化される。そして、スケールファクタを符号化する方式は、一層効率の良い圧縮のために僅かに変形された方式を使用する。

【0032】まず、スケールファクタの符号化について説明する。本実施形態によるステレオ信号を処理する符号化装置は、符号化の効率を一層高めるために、2種類の方法を利用してスケールファクタを符号化する。符号化装置は、前記2種類の方法のうちの性能が良好なものを選択して復号化装置へ伝送する。

【0033】第1の方法は、スケールファクタを圧縮するために、まずスケールファクタ値のうちの最大値max scalefactor を求める。そして、各スケールファクタと最大値max scalefactor に対する差分を求める。前記スケールファクタ差分値を算術符号化する。このために、表5、5ないし表5、8に示す4種類のモデルを使用する。前記モデルに関する情報は、scalefactor model に記憶される。

【0034】

【表1】

【表5.5】スケールファクタ差分値に対する算術モデル1

寸法 (size)	累積頻度数 (cumulative frequencies)
8	1342, 790, 510, 344, 214, 127, 57, 0

【0035】

【表2】

【表5.6】スケールファクタ差分値に対する算術モデル2

寸法 (size)	累積頻度数 (cumulative frequencies)
16	2441, 2094, 1798, 1563, 1347, 1154, 966, 818, 634, 464, 342, 241, 157, 97, 53, 0

【0036】

【表3】

【表5.7】スケールファクタ差分値に対する算術モデル3

寸法 (size)	累積頻度数 (cumulative frequencies)
32	3963, 3525, 3188, 2949, 2705, 2502, 2286, 2085, 1868, 1668, 1516, 1354, 1207, 1055, 930, 821, 651, 510, 373, 269, 192, 134, 90, 68, 37, 29, 24, 15, 10, 8, 5, 0

【0037】

【表4】

[表 6.8] スケールファクタ差分値に対する算術モデル 4

寸法 (size)	累積頻度数 (cumulative frequencies)
64	12687, 13282, 12961, 12656, 12165, 11721,
	11250, 10582, 10042, 9587, 8742, 8010, 7256,
	6619, 6042, 5480, 4898, 4331, 3817, 3374, 3058,
	2759, 2545, 2383, 2192, 1989, 1812, 1682, 1390,
	1163, 1037, 935, 668, 518, 438, 358, 245, 187, 181,
	149, 144, 128, 122, 117, 112, 106, 101, 85, 80, 74,
	69, 64, 58, 53, 48, 42, 37, 32, 26, 21, 16, 10, 5, 0

【0038】第2の方法は、スケールファクタを圧縮するために、第1の方法と同様に、まず、スケールファクタ値のうちの最大値max scalefactor を求める。そして、最初のスケールファクタと最大値max scalefactor との差分を求め、この差分値を算術符号化する。そして、残りのスケールファクタは以前のスケールファクタとの差分を求め、その差分値を算術符号化する。この場合は、使用されるモデルが決まっているので、最大値scalefactor model 値は役割をしない。

【0039】次に、ステレオ信号から量子化した周波数成分に対する符号化について説明する。各チャンネルに対する量子化したデータをビット分割する。モノラルチャンネルに対する信号を処理するときは、ビット分割されたデータを4次元ベクトルで結び付け、この4次元ベクトルを符号化の基本単位として使用する。ステレオチャンネルの信号を符号化するときも、これらの点はそのまま維持する。即ち、重要度が最高のビットに対する符号化を始めることになる。ビット分割されたデータの4次元ベクトルを、まず左側のチャンネルに対して算術符号化する。次に、同じ周波数位置の右側のチャンネルに対する4次元ベクトルを算術符号化する。このように、左側のチャンネルと右側チャンネルとを引き続き組み込みながら符号化を行う。チャンネルが1つとすれば、MSBからLSBの順に符号化する。そして、同じ重要度のビット分割されたデータの符号化は、低い周波数から高い周波数の順に進む。このとき、各ベクトルに割当てられたビットが現在符号化中のビットの重要度よりも高いとすれば、このベクトルは符号化する必要がないことを意味するので、そのまま通過する。

XQ0, XQ1, XQ2, XQ3, XQ4, XQ5, ..., XQk, ...

ここで、XQkは、4\*kから(4\*k+3)番目までの量子化した周波数成分のビット分割されたデータである。

【0040】チャンネルが2つの場合も、同様にMSBからLSBの順に符号化する。同じ重要度のビット分割されたデータの符号化は、低い周波数から高い周波数の順に進む。しかし、チャンネル数が2つであることに鑑みて符号化の順番を決めることになる。即ち、左側のチャネ

ルと右側のチャンネルに量子化した周波数成分が次の通りであるとすれば、

左側のチャンネル：XQL0, XQL1, XQL2, XQL3, XQL4, XQL5, ..., XQLk, ...

右側のチャンネル：XQR0, XQR1, XQR2, XQR3, XQR4, XQR5, ..., XQRk, ...

但し、XQLk, XQRkは、4\*kから(4\*k+3)番目までの量子化した周波数成分のビット分割されたデータである。このように、チャンネルが2つの場合にも、チャンネルが1つの場合と同様に、低い周波数から高い周波数の順に進む。しかし、チャンネル同士で重要な成分を先立って符号化するため、チャンネル成分間にはインターリーブが施される。即ち、各ベクトルに対して符号化が進む順番は、2つのチャンネル間に次のように交互に出されることになる。

XQL1, XQR1, XQL2, XQR2, ...

このように構成された情報は、2つのチャンネルを統合して重要な信号の順に符号化が進むので、スケラブルオーディオコーデックにビット率が減っても、性能の低下は顕著ではない。

【0041】ここで、本発明の望ましい実施形態について説明する。本実施形態では、符号化効率を高めるための付加的なモジュール等、全てのモジュールをきむAACの基本構造に適用され、ビット率の調節可能なデジタルオーディオデータ符号化装置を提供する。即ち、AACの符号化/復号化に使用される基本モジュールはそのまま使用し、無損失符号化モジュールのみをビット分割符号化方式に取替え、ビット率の調節可能な符号化器を提供する。1本のビットストリーム内に1つのビット率に対するもののみで構成するのではなく、基本階層を基に多数の高位階層のデータをも共に表現する。従って、多数の階層のビット率に関する情報を1本のビットストリームで表現するに当り、図2に示すように、階層的な構造でオーディオ信号を符号化するが、但し、重要な信号成分を優先的に表現する。

【0042】本実施形態では、BSACスケラブルコーデックの無損失符号化過程の前まではAACと同様のモジュールを使用する。従って、AACビットストリームを復号化して量子化した周波数データを作ると、この

復号化したデータを再びBSACスケーラブルビットストリームで作ることができる。これは、AACビットストリームとBSACスケーラブルビットストリームとの間には無損失変号化（lossless transcoding）が可能であるとの意味である。結局、環境や状況に応じて適合するビットストリームの形に互いに変換可能である。従って、符号化効率とスケーラビリティ機能とを同時に満足すると共に、相互補完的な関係維持が可能であり、他のスケーラブルコーデックとは差別性を有する。

【0043】このように作られたビットストリームは、使用者の要求、又は伝送経路の状態によって最高のビット率をもつビットストリームに含まれた低いビット率のビットストリームを簡単に再構成することにより、低いビット率のビットストリームを作ることができる。即ち、符号化器で作られたビットストリームや、又は或る媒体に保存されているビットストリームを使用者の要求に応じて所望のビット率に対するビットストリームにリアルタイムで作って伝送することができる。又、使用者が完全なビットストリームを有しているとしても、使用者のハードウェアの性能が良くないとか、使用者が復号化器の複雑性を低めようとする場合、このビットストリ

〔表 2.1〕各階層に対するビット率（8kbps 間隔）

階層 (layer)	ビット率 (bitrate)
0	16kbps
1	24kbps
2	32kbps
3	40kbps
4	48kbps
5	56kbps
6	64kbps

【0046】一方、一層精密な間隔で階層を構成することもできる。基本階層は16kbps、最高位階層が64kbpsであり、各階層を1kbps間隔のビット率をもつビットストリームから構成することができる。各階層は表3.1のように構成できる。16kbpsから64kbpsまで、1kbps間隔でスケール調節が可

能なビットストリームを構成するファイングラニュールスケーラビリティ（fine granule scalability）を具現することもできる。

【0044】一例として、基本階層は16kbps、最高位階層は64kbpsであり、各階層を8kbps間隔のビット率を有するビットストリームから構成することができる。即ち、16、24、32、40、48、56、64kbpsの7階層のスケール調節が可能なビットストリームを構成する例である。各階層は、表2.1のように定義できる。符号化装置で作られるビットストリームは、図2に示すような階層的構造を有するので、最高位階層の64kbpsに対するビットストリーム内に各階層（16、24、32、40、48、56、64kbps）に対するビットストリームが含まれている。もし、或る使用者が最高位階層に対するデータを要求すれば、このビットストリームは加工無しに送られる。又、他の使用者が基本階層（16bpsに相当する）に対するデータを要求すれば、単に前の部分のビットストリームのみが切出されて送られる。

【0045】

〔表 5〕

能なビットストリームを構成するファイングラニュールスケーラビリティ（fine granule scalability）を具現することもできる。

【0047】

〔表 6〕



【表 3.1】 各階層に対するビット率 (1kbps 間隔)

階層	ビット率	階層	ビット率	階層	ビット率	階層	ビット率
0	16kbps	12	28kbps	24	40kbps	36	52kbps
1	17kbps	13	29kbps	25	41kbps	37	53kbps
2	18kbps	14	30kbps	26	42kbps	38	54kbps
3	19kbps	15	31kbps	27	43kbps	39	55kbps
4	20kbps	16	32kbps	28	44kbps	40	56kbps
5	21kbps	17	33kbps	29	45kbps	41	57kbps
6	22kbps	18	34kbps	30	46kbps	42	58kbps
7	23kbps	19	35kbps	31	47kbps	43	59kbps
8	24kbps	20	36kbps	32	48kbps	44	60kbps
9	25kbps	21	37kbps	33	49kbps	45	61kbps
10	26kbps	22	38kbps	34	50kbps	46	62kbps
11	27kbps	23	39kbps	35	51kbps	47	63kbps
						48	64kbps

【0048】 各階層は、ビット率に応じて帯域幅が制限されるようにするが、8kbps 間隔でスケラビリティを提供しようとするば、帯域制限は表 2. 2 と表 2. 3 に示す通りである、1kbps 間隔であれば、帯域制

限は表 3. 2 と表 3. 3 のように定義される。

【0049】

【表 7】

【表 2.2】 短いウィンドウに対する各階層での帯域制限 (8kbps 間隔)

階層 (layer)	帯域制限 (band limit)
0	20
1	28
2	40
3	52
4	60
5	72
6	84

【0050】

【表 8】

【表 2.3】 長いウィンドウ に対する各階層での帯域制限 (8kbps 間隔)

階層 (layer)	帯域制限 (band limit)
0	160
1	244
2	328
3	416
4	500
5	584
6	672

【0051】

【表 9】

【表 3.2】 短いウィンドウに対する各階層での帯域制限 (1kbps 間隔)

階層	帯域制限	階層	帯域制限	階層	帯域制限	階層	帯域制限
0	20	12	36	34	62	26	68
1	20	13	36	25	62	27	68
2	20	14	36	26	52	38	68
3	24	15	40	27	56	39	72
4	24	16	40	28	56	40	72
5	24	17	40	29	56	41	72
6	28	18	44	30	60	42	76
7	28	19	44	31	60	43	76
8	28	20	44	32	60	44	76
9	32	21	48	33	64	45	80
10	32	22	48	34	64	46	80
11	32	23	48	35	64	47	80
						48	84

【0052】

【表 10】

【表 3.3】 長いウィンドウに対する各階層での帯域制限 (1kbps 間隔)

階層	帯域制限	階層	帯域制限	階層	帯域制限	階層	帯域制限
0	160	12	288	24	416	36	544
1	168	13	296	25	424	37	552
2	160	14	304	26	436	38	564
3	192	15	320	27	448	39	576
4	200	16	328	28	456	40	584
5	212	17	340	29	468	41	596
6	224	18	352	30	480	42	608
7	232	19	360	31	488	43	616
8	244	20	372	32	500	44	628
9	256	21	384	33	512	45	640
10	264	22	392	34	520	46	648
11	276	23	404	35	532	47	660
						48	672

【0053】 入力データは、48kHzにサンプリングされたPCMデータであり、1フレームの大きさは1024個である。64kbpsのビット率の場合、1フレームで使えるビットの数は平均64000ビット/秒\*(1024/48000)=1365.3333ビットになる。同様に、各ビット率によって1フレームに使えるビットの大きさを計算する

ことができる。このように計算されたフレーム当りの使用可能なビット数は、8kbps間隔のときは表2、4の通りであり、1kbps間隔のときは表3、4の通りである。

【0054】

【表 11】

【表 2.4】各階層でのチャンネル当り利用可能なビット (8kbps 間隔)

階層 (layer)	利用可能ビット数
0	341
1	512
2	682
3	853
4	1024
5	1194
6	1365

【0065】

【表 12】

【表 3.4】各階層でのチャンネル当り利用可能なビット (8kbps 間隔)

階層 (layer)	利用可能なビット数	階層 (layer)	利用可能なビット数	階層 (layer)	利用可能なビット数	階層 (layer)	利用可能なビット数
0	341	12	697	24	858	36	1109
1	512	13	618	25	874	37	1180
2	682	14	640	26	896	38	1162
3	853	15	661	27	917	39	1173
4	1024	16	682	28	938	40	1194
5	1194	17	704	29	960	41	1216
6	1365	18	725	30	981	42	1237
7	1536	19	746	31	1002	43	1258
8	1707	20	768	32	1024	44	1280
9	1878	21	789	33	1045	45	1301
10	2049	22	810	34	1066	46	1322
11	2220	23	832	35	1088	47	1344
						48	1365

【0066】ここで、本実施形態によるステレオオーディオ信号の符号化過程及び復号化過程を更に具体的に説明する。

#### 1. 符号化過程

全体的な符号化過程は、MPEG-2 AAC国際標準案に記載された過程を経た後、無損失符号化過程は、本発明で提案したビット分割符号化を適用する。

##### (1. 1) 音響心理部

まず、入力データを受入れ、音響心理モデル (Psychoacoustic Model) を用い、今処理されているフレームのブロック形態 (long, start, short, stop)、各スケールファクタバンドのSMR (Signal-to-Masked Threshold Ratio) 値、短いブロック (short block) の場合のグループ情報、そして音響心理モデルと時間/周波数の同期を合わせるために時間遅延されたPCMデータ等を作って時間/周波数マッピング部100へ送る。音響心理モデルを計算する方法は、次のISO/IEC 11172-3 のモデル2を使用する。

MPEG Committee ISO/IEC/JTC1/SC29/WG11, Information technology-Coding of moving pictures and associated

audio for data storage mediated about 1.5Mbit/s-Part 3: Audio, ISO/IEC IS 11172-3, 1993

このモジュールは必ず使用すべきであるが、使用者毎に他のモデルを使用することもある。

【0067】(1. 2) 時間/周波数マッピング部  
時間/周波数マッピングは、MPEG-2 AAC国際標準案に定義されているものを使用する。音響心理モデルの出力であるブロック形態に応じて、時間/周波数マッピング部100では、MDCTを利用して時間領域のデータを周波数領域のデータに切替える。このとき、ブロックの大きさはlong/start/stop ブロックの場合に2048であり、短いブロックの場合に256であり、各ブロックに対してMDCTを8回行う。そして、ビットストリーム形成部180にウィンドウタイプ(window type)、ウィンドウグループ情報等を伝達する。ここまでの過程は既存のMPEG-2 AAC (MPEG Committee ISO/IEC/JTC1/SC29/WG11, ISO/IEC MPEG-2 AAC IS 13818-7, 1997) で使用されるものと同様の方式を使用する。

【0068】(1. 3) 時間領域雑音形状化 (TNS; Temporal Noise Shaping) 部

TNSは、MPEG-2 AAC国際標準案に定義されているものを使用する。時間領域雑音形状化部120は選択モジュールであり、変換の各ウィンドウ内で量子化雑音の時間的な形状を制御する。周波数データのフィルタリング過程を適用することにより、時間領域雑音形状化が可能である。そして、ビットストリーム形成部180にTNS情報を伝達する。

【0069】(1.4) インテンシティストereo処理部

インテンシティストereo処理部130は、MPEG-2 AAC国際標準案に定義されているものを使用する。インテンシティストereo処理部130は、ステレオ信号を一層効率良く処理するものの1つであり、2つのチャンネルのうちのいずれか一方のチャンネルに対するスケールファクタバンドに関する量子化した情報のみを符号化し、他方のチャンネルはスケールファクタのみを伝送する。このモジュールは選択モジュールであり、符号化器で色々な事項を考慮して各スケールファクタバンド単位で使用可否を判断することができる。そして、ビットストリーム形成部180にインテンシティストereoフラグ値を伝達する。

【0060】(1.5) 予測部

予測部140は、MPEG-2 AAC国際標準案に定義されているものを使用する。予測部140は選択モジュールであり、現フレームの周波数係数の値を予測する。そして、ビットストリーム形成部180に予測に係わっ

たパラメータを伝達する。

【0061】(1.6) M/Sステレオ処理部

M/Sステレオ処理部150は、MPEG-2 AAC国際標準案に定義されているものを使用する。M/Sステレオ処理部150は選択モジュールであり、ステレオ信号を一層効率良く処理するものの1つであり、左チャンネルの信号と右チャンネルの信号とをそれぞれ和した和信号と引いた差信号とに切替えた後、この信号を処理するものである。各スケールファクタバンド単位で使用可否を判断することができる。

【0062】(1.7) 量子化部

周波数領域に切替えられたデータは、表1.1又は表1.2に示すように、スケールファクタバンドで周波数成分を結び付け、スケールファクタバンドのSNR (Signal-to-Noise Ratio) 値が音響心理モデルの出力値であるSMR値よりも小さくなるようにスケールファクタを変化させて量子化を行う。量子化は、スカラー量子化 (scala quantization) からなり、スケールファクタの間隔は $2^{1/4}$ を用いる。量子化は、SNR値とSMR値とを考慮して人間が感じる雑音の量を最小化する。正確な量子化過程は、MPEG-2 AACに記述されている過程を使用する。このとき、得られる出力は、量子化したデータと各スケールファクタバンドのスケールファクタである。

【0063】

【表1.3】長いブロックに対するスケールファクタバンド

Swb	swb_offset_ long_window	swb	swb_offset_ long_window	swb	swb_offset_ long_window	swb	swb_offset_ long_window
0	0	12	56	24	196	37	576
1	4	18	64	26	216	38	608
2	8	14	72	26	240	39	640
3	12	15	80	27	264	40	672
4	16	16	88	28	292	41	704
5	20	17	96	29	320	42	736
6	24	18	108	30	352	43	768
7	28	19	120	31	384	44	800
8	32	20	132	32	416	45	832
9	36	21	144	33	448	46	864
10	40	22	160	34	480	47	896
11	48	23	176	35	512	48	928
				36	544		1024

【0064】

【表1.4】

【表 1.2】 短いブロックに対するスケールファクタバンド

awb	awb_offset_ long_window	awb	awb_offset_ long_window
0	0	8	44
1	4	9	56
2	8	10	68
3	12	11	80
4	16	12	96
5	20	13	112
6	28		128
7	36		

【0065】 (1. 8) ビット分割算術符号化(Bit-sliced Arithmetic Coding)を用いたビットパッキング  
ビットパッキングは、ビット分割算術符号化部170とビットストリーム形成部180とを介して行われ、周波数成分を符号化の便利性的のために再配列する。ブロック形態によって再配列される順番が異なる。ブロック形態が長いウィンドウを使用する場合、図4に示すように、スケールファクタバンドの順に配列される。ブロック形態が短いウィンドウの場合は、図6に示すように、8個の各ブロックの周波数成分のうちから4個ずつ昇降順に繰返して挟み込まれて配列される。

【0066】 このように再配列された量子化したデータとスケールファクタとを、階層構造をもったビットストリームから構成することになる。ビットストリームは、表7. 1ないし表7. 11のように、シンタキストを以て構成される。ビットストリームの構成要素は、前方にAACと共に使用できる部分は共有し、新しく本発明を適用するためのものは別に分離した。従って、基本的な構造はAAC標準案のように構成した。

【0067】

【表 1.6】

【表 7.1】 Syntax of bsac\_lstep\_data\_block ()

Syntax	No. of bits	Mnemonics
<pre> bsac_lstep_data_block() {     lslayer=0;     while(data_available() ) {         bsac_lstep_stream(lslayer)         lslayer++;     } } </pre>		

【0068】

【表 1.6】

【表 7.2】 Syntax of bsac\_lstep\_stream()

Syntax	No. of bits	Mnemonics
<pre> bsac_lstep_stream(lslayer) {     for(k=lsstep_offset[lslayer]; k&lt;lsstep_offset[lslayer]; k++)         BSAC_stream_buf[k] /* Large step stream is saved in BSAC_stream_buf []    BSAC_stream_buf[] is mapped to small step stream,    bsac_raw_data_block(), for the actual decoding.    see the decoding process of BSAC large step    scalability    for more detailed description. */ } </pre>	8	unimabf

[0069]

[表 17]

[表 7.3] Syntax of bsac\_raw\_data\_block()

Syntax	No. of bits	Mnemonics
<pre>bsac_raw_data_block() {     bsac_main_stream()     layer=1;     while(data_available() &amp;&amp; layer &lt;=encoded_layer)     {         bsac_layer_stream(nch, layer)         layer++;     }     byte_alignment() }</pre>		

[0070]

[表 18]

[表 7.4] Syntax bsac\_main\_stream()

Syntax	No. of bits	Mnemonics
<pre>bsac_main_stream {     nch     switch(nch) {         case 1: bsac_single_main_stream()             break         case 2: bsac_pair_main_stream()             break     } }</pre>	3	uimabf

[0071]

[表 19]

[表 7.5] Syntax of bsac\_single\_main\_stream()

Syntax	No. of bits	Mnemonics
<pre>bsac_single_main_stream() {     ltp_data_present     if(ltp_data_present)         ltp_data()     bsac_channel_stream(1,1) }</pre>	1	uimabf

[0072]

[表 20]

**[表 7.6] Syntax of bsac\_pair\_main\_stream()**

Syntax	No. of bits	Mnemonic
<pre> testc_pair_main_stream() {     ltp_data_present     if(ltp_data_present) {         ltp_data()         ltp_data()     }     common_window     if(common_window)         stereo_mode         bsac_channel_stream(2, common_window) </pre>	<p>1</p> <p>1</p> <p>2</p>	<p>uimshf</p> <p>uimshf</p> <p>uimshf</p>

【表 2 1】

**[表 7.7] Syntax of `base_layer_stream0`**

Syntax	No. of bits	Mnemonics
<pre> baac_layer_stream(nch, layer) {     baac_side_info(nch, layer)     baac_spectral_data(nch, layer) } </pre>		

【表 2 2】

**[表 7.8] Syntax of bsac\_channel\_stream()**

Syntax	No. of bits	Meaning
bsac_channel_stream(nch, common_window)		
{		
for(ch=0; ch<nch; ch++)		
max_scalefactor[ch]	8	uimbf
ica_info()		
if(!common_window)		
ica_info()		
for(ch=0; ch<nch; ch++) {		
tns_data_present[ch]	1	cimbf
if(!tns_data_present[ch])		
tns_data()		
gain_control_data_present[ch]	1	uimbf
if(gain_control_data_present[ch])		
gain_control_data()		
}		
PNS_data_present	1	uimbf
if(PNS_data_present)		
PNS_start_sfb	8	uimbf
bsac_general_info(nch)		
bsac_layer_stream(nch, 0)		

【表 2 3】

【表 7.9】 Syntax of base\_general\_info()

Syntax	No. of bits	Mnemonics
base_general_info(nch)		
{		
frame_length	10/11	uimbf
encoded_layer	6	uimbf
for(ch=0;ch<nch;ch++) {		
scalefactor_model[ch]	2	uimbf
min_ArModel[ch]	5	uimbf
ArModel_model[ch]	2	uimbf
ecf_coding[ch]	1	uimbf
}		

【0076】

【表 2 4】

【表 7.10】 Syntax of base\_side\_info()

Syntax	No. of bits	Mnemonics
base_side_info(nch, layer)		
{		
if(ch==1 && PNS_data_present) {		
for(nf=PNS_start_nf; nf<max_nf; nf++)		
encode_noise_flag[g][nf]	0..1	bool
1 bit		
}		
else if(stereo_mode==1    PNS_data_present) {		
for(g=0;g<num_sideinfo_group;g++)		
for(nf=layer_nf[layer]; nf<layer_nf[layer+1]; nf++)		
if(stereo_mode==0)		
encode_noise_flag[g][nf]	0..1	bool
else if(stereo_mode==3) {		
encode_stereo_info[g][nf]	8..2	bool
}		
if(PNS_data_present && nf==PNS_start_nf) {		
if(stereo_info==0    stereo_info==1)		
encode_noise_flag[g][nf]	0..1	bool
encode_noise_flag_r[g][nf]	0..1	bool
}		
if(stereo_info==3) {		
if(noise_flag_r[g][nf] && noise_flag_r[g][nf])		
encode_stereo_mode[g][nf]	0..8	bool
}		
}		
for(ch=0;ch<nch;ch++)		
for(g=num_sideinfo_group;g++)		
for(nf=layer_nf[layer]; nf<layer_nf[layer+1]; nf++)		
encode_noise_flag[g][nf]	0..13	bool
}		
for(ch=0;ch<nch;ch++)		
for(nf=layer_nf[layer]; nf<layer_nf[layer+1]; nf++)		
for(g=num_sideinfo_group;g++) {		
band=(nf>num_sideinfo_group)?g		
for(nf=0;nf<band;nf++) {		
if(band==1; 1-4) {		
channel=ntt2ch(ch, 1);		
if(decode_channel[ch][band])		
encode_ArModel[ch][band]	0..13	bool
decode_channel[ch][band]=1;		
}		
}		
}		

【0077】

【表 2 5】



【表 7.11】 Syntax of bsac\_spectral\_data()

Syntax	No. of bits	Comments
<pre> bsac_spectral_data(nch, layer) {     for(snf=maxsnf; snf&gt;0; snf--) {         for(i=0; i&lt;last_index; i +=4) {             for(ch=0; ch&lt;nch; ch++) {                 if(i&gt;=layer_index[ch]) continue;                 if(cur_snf[ch][i]&lt;snf) continue;                  dim0 = dim1 = 0                 for(k=0; k&lt;4; k++)                     if(prestate[ch][i+k]) dim1++                     else dim0++                  if(dim0)                     acode_voc0                 if(dim1)                     acode_voc1                  for(k=0; k&lt;4; k++) {                     if(sample[ch][i-k] &amp;&amp; !prestate[ch][i+k]) {                         acode_sign                         prestate[ch][i+k]=1                     }                 }                 cur_snf[ch][i]--                 if(total_estimated_bits&gt;=available_bits[layer])return             }         }         if(total_estimated_bits&gt;=available_bits[layer])return     } } </pre>	0.14	balbf
	0.14	balbf
	0.1	balbf

【0078】一方、本発明で新しく使用したビットストリームの構成要素について説明する。

(1. 8. 1) bsac channel streamの符号化  
common window: 2チャンネルが同じ形態のブロックを使用するか否かを表す。

max scalefactor ch : スケールファクタのうちの最大値、整数値で8ビット

tns\_data\_present ch : TNSが符号化器で使用されたか否かを表す。

gain\_control\_data\_present ch : AACでスケラブルサンプリングレート (SSR) プロファイルを支援するための時間/周波数マッピング方法が使用されたことを表すフラグ

stereo\_mode: ステレオ処理方法を表す2ビットフラグ

- 00 Independent
- 01 All ms used are ones
- 10 1bit mask of max sfb bands of ms used is located in the layer side information part.
- 11 2bit mask of max sfb bands of stereo info is located in the layer side information part.

【0079】(1. 8. 2) bsac\_dataの符号化  
frame\_length: 1枚のフレームに対するビットストリーム全体に対する大きさ、byte単位に表示される。モノラル信号の場合9ビット、ステレオ信号の場合10ビットを使用する。

encoded\_layer: ビットストリームの符号化した最高位階層に関する情報を符号化する。8kbp s間隔であれば

3ビット、1kbp s間隔であれば6ビットが使用される。階層に関する情報は表2. 1又は表3. 1に示す。  
scalefactor\_model ch : スケールファクタの差を算術符号化する時に使用するモデルに関する情報であり、表4. 2に示す。

【0080】

【表26】

【表 4.2】 Arithmetic Model of Differential Scalefactor

Model Number	Largest Differential Scalefactor	Model listed in Table
0	7	5.5
1	16	5.6
2	31	5.7
3	63	5.8

【0081】min\_ArModel ch : 算術符号化モデル指数のうちの最小値

ArModel\_model ch : 算術符号化モデル指数(ArModel)とmin\_ArModelとの差分信号を算術符号化するために

使用されるモデルに関する情報であり、表 4. 3 に示す。

【0082】

【表 27】

【表 4.3】 Arithmetic Model of Differential ArModel

Model Number	Largest Differential Scalefactor	Model listed in Table
0	3	5.9
1	7	6.10
2	15	6.11
3	31	6.12

【0083】

【表 28】

【表 5.8】 differential ArModel arithmetic model 1

寸法 (size)	累積頻度数 (cumulative frequencies)
4	9868, 3351, 1675, 0,

【0084】

【表 29】

【表 5.10】 differential ArModel arithmetic model 2

寸法 (size)	累積頻度数 (cumulative frequencies)
8	12492, 8600, 5941, 3282, 2155, 1028, 514, 0,

【0085】

【表 30】

【表 5.11】 differential ArModel arithmetic model 3

寸法 (size)	累積頻度数 (cumulative frequencies)
16	14316, 12348, 9882, 7516, 6399, 5282, 4183, 3083, 2247, 1411, 860, 309, 185, 61, 31, 0,

【0086】

【表 31】

【表 5.12】 differential ArModel arithmetic model 4

寸法 (size)	累積頻度数 (cumulative frequencies)
40	12170, 7966, 6429, 4901, 4094, 3287, 2982, 2677, 2454, 2230, 2062, 1894, 1621, 1348, 1199, 1050, 854, 638, 468, 278, 189, 59, 38, 18, 17, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0,

【0087】 (1. 8. 3) bsac side infoの符号化

このように、全階層に共通して使える情報を以てまずビットストリームを符号化し、次に各階層に共通して使用される付加情報を符号化する。

acode ms used [g sfb :window group g, scalefactor band scf で M/S 符号化が使用されたか否かを表す 1 ビットフラグの ms used を算術符号化してできたコードワード。ms used は以下のように定義される。

0 Independent

1 ms used

acode stereo info g sfb :window group g, scalefactor band scf で M/S やインテンシティステレオ符号化が使用されたか否かを表す 2 ビットフラグの stereo info を算術符号化してできたコードワード。stereo info は以下のように定義される。

00 Independent

01 ms used

10 intensity in phase

11 intensity out of phase

acode scf : スケールファクタを算術符号化してできたコードワード。

acode ArModel : ArModel (モデル指数) を算術符号化してできたコードワード。モデル指数は表 4. 3 に示されたモデルのうちのいずれが選択されたかを表す情報である。

【0088】 (1. 8. 4) bsac spectral data の符号化

各階層に共通して使用される付加情報を使用した後、量子化した周波数成分を BSAC (ビット分割算術符号化) 技術を用いてビット分割した後に算術符号化する。

acode vec0 : 1 番目のサブベクトルである subvector0

をモデル指数値として決まった算術モデルを以て算術符号化した結果のコードワード。

acode vac1: 2番目のサブベクトルであるsubvector1をモデル指数値として決まった算術モデルを以て算術符号化した結果のコードワード。

acode sign: 符号ビットに関する情報を表6. 15のモデルを以て算術符号化した結果のコードワード。

【0089】

【表32】

【表5.15】 sign arithmetic model

寸法 (size)	累積頻度数(Cumulative frequencies)
2	8182, 0

【0090】各サブベクトルを符号化しながら使用されたビット数を計算し、各階層で使用可能なビット数と比較しつつ、使用されたビット数が使用可能なビット数に等しいかそれともこれを上回るとき、次階層の符号化を改めて始める。基本階層の帯域幅は、長いブロックの場合、21番目のスケールファクタバンドまで帯域制限される。そこで、21番目のスケールファクタバンドまでのスケールファクタと相応する各符号化帯域の算術符号化モデルを符号化する。算術符号化モデルから各符号化帯域のビット割当情報を得ることができ、これらの値のうちの割当されたビットの最大値を求め、この最大値から前述の方法によって符号化を始める。そして、次第に次のビットに対する符号化を進める。又、符号化中のビットよりも或る符号化帯域の割当ビットが小さい場合、符号化せずにそのまま通過し、該符号化帯域は、割当ビットと符号化中のビットとが等しくなってから初めて符号化を行う。基本階層のビット率は16kbpsであるから、許容可能な全体ビットは336ビットである。従って、使用される総ビット量を計算し続けてからビット量が336ビット以上になった場合、符号化を止める。

【0091】基本階層(16kbps)に対するビットストリームを作り切った場合、その次の階層に対してビットストリームを作成する。次の階層に対する符号化は、制限される帯域が大になる。従って、基本階層において制限される帯域に比べて新しく追加される帯域に対してのみスケールファクタと算術符号化モデルに対する符号化を行う。そして、基本階層において各帯域の符号化できずに残っているビット分割されたデータと、新しく追加された帯域のビット分割されたデータとのうち、重要度が高いものから基本階層での方式と同様にして符号化を進めながら、使用された総ビット量の大きさが使用可能なビット数よりも大きくなった場合、符号化過程を止めて次の階層のビットストリーム作成の用意をする。この方法により、残りの階層に対するビットストリームを作成することができる。

【0092】2. 復号化過程

(2. 1) ビットストリームの分析及び復号化

(2. 1. 1) bsac channel streamの復号化

bsac channel streamにおいて、復号化順序は、まず max scalefactor を得る。それから ics info () を得、TNSデータが存在すればTNSデータを得る。そして、チャンネルが2つであれば、stereo mode値を得た後、BSACデータを得る。

(2. 1. 2) bsac dataの復号化

frame length、encoded layer、スケールファクタ及び算術符号化モデルを復号化するに必要な付加情報をビットストリームで復号化する。

(2. 1. 3) BSACストリームの復号化

BSACストリームは、階層構造を有している。まず、基本階層に関する付加情報を分離し、算術復号化を行う。次に、量子化した周波数成分のビット分割データを分離して算術復号化する。そして、その次の階層に関する付加情報を復号化し、量子化した周波数成分のビット分割データを算術復号化する。

【0093】これらの各階層に関する付加情報復号化とビット分割データの復号化過程とは、階層が符号化した階層よりも大きくなるまで繰返される。

(2. 1. 4) stereo info又はms usedの復号化

stereo info又はms usedの復号化は、ステレオマスクを表すstereo modeによって左右される。stereo modeが0又は1であれば、stereo info又はms usedに対する算術復号化が不要である。stereo modeが1であれば、全てのms used値が1になる。このms used情報はM/Sステレオ処理部150に伝達され、M/Sステレオ処理が施される。stereo modeが2であれば、ms used値をスケールファクタバンド単位を表5. 13に示されたモデルを用いて算術復号化する。このms used情報がM/Sステレオ処理部150に伝達され、M/Sステレオ処理が施される。

【0094】

【表33】

【表5.13】 MS\_used model

寸法 (size)	累積頻度数 (cumulative frequencies)
2	11469, 0

stereo modeが3であれば、表5. 14に示されたモデルを用いてstereo infoが算術復号化される。このように復号化したデータがM/Sステレオ処理部150又はインテンシティステレオ処理部130に伝達され、各ス

【表 5.14】 stereo\_info model

寸法 (size)	累積頻度数 (cumulative frequencies)
4	13926, 4096, 1638, 0

【0096】(2. 1. 5) bsac\_side\_infoの復号化

以上で作られたスケーラブルビットストリームは、階層構造を有している。まず、基本階層に関する付加情報をビットストリームから分離して復号化する。そして、基本階層のビットストリームに含まれる量子化した周波数成分のビット分割情報をビットストリームから分離して復号化する。他の高位階層に対しても基本階層と同様の過程を経て復号化をすることができる。

【0097】(2. 1. 5. 1) スケールファクタの復号化

周波数成分は、4の倍数の周波数係数を含むスケールファクタバンドに分けられる。各スケールファクタバンドは、それぞれ1つのスケールファクタを有する。スケールファクタを復号化する方法は、2種類ある。このうち、いずれの方法により復号化を行うかについては、s\_cfcoding値に貯蔵されている。

【0098】第1番目の方法は、max\_scalefactorを8ビットの符号のない整数に復号化する。一般に、符号

ケルファクタバンド単位にAACに記されたM/S又はインテンシティステレオ処理が施される。

【0095】

【表 3.4】

化を行うとき、スケールファクタの差分をマッピングした値が符号化される。従って、各スケールファクタバンドに対してこのマッピングした値を表5. 2のモデルを用いて算術復号化する。このとき、算術復号化した値が54であれば、マッピングされた値が54以上の値になるので、54とこの値との差分が更に符号化されており、この値を更に復号化して54以上の値に復元する。このように、マッピングした値に対する復号化が完了すれば、このマッピングされた値を差分信号に逆マッピングすることになる。これらのマッピングと逆マッピングは、表5. 1と表5. 2のマッピングテーブルに基づいて行われる。最初のスケールファクタに対しては、max\_scalefactor との差分信号を以てスケールファクタを求めることができる。そして、残りのスケールファクタに対しては、差分と以前のスケールファクタを和して元のスケールファクタを復元できる。この時、使用される確率モデルは表5. 3及び表5. 4である。

【0099】

【表 3.5】

【表 3.1】 differential scalefactor to index transition table

Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index
0	68	16	87	32	46	48	35	64	9	80	40	98	96	118	112
1	69	17	88	33	47	49	19	66	10	81	43	97	97	119	113
2	70	18	89	34	48	50	20	68	12	82	44	98	98	114	114
3	71	19	90	35	49	51	14	67	13	83	45	99	99	116	116
4	72	20	91	36	50	52	15	68	17	84	52	100	100	118	118
5	73	21	92	37	51	53	16	69	18	85	53	101	101	117	117
6	74	22	93	38	52	54	11	70	21	86	63	102	102	118	118
7	75	23	94	39	53	55	7	71	22	87	66	103	103	119	119
8	76	24	95	40	54	56	8	72	26	88	84	104	104	120	120
9	77	25	96	41	55	57	6	73	27	89	57	105	105	121	121
10	78	26	97	42	56	58	3	74	28	90	74	106	106	122	122
11	79	27	98	43	57	59	1	75	31	91	81	107	107	123	123
12	80	28	99	44	58	60	0	76	32	92	82	108	108	124	124
13	81	29	100	45	59	61	3	77	33	93	83	109	109	126	126
14	82	30	101	46	60	62	4	78	34	94	94	110	110	128	128
15	83	31	102	47	61	63	5	79	39	95	95	111	111	127	127

【0100】

【表 3.6】

【表 5.2】 index to differential scalefactor transition table

index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff	index	Diff
0	60	16	38	32	76	48	34	64	56	50	9	96	96	112	112
1	69	17	68	33	77	49	35	65	22	81	10	97	97	113	113
2	53	18	69	34	78	50	36	66	23	82	11	98	98	114	114
3	61	19	49	35	40	81	37	67	25	83	12	99	99	115	115
4	62	20	50	36	41	52	64	38	0	84	13	100	100	116	116
5	67	21	70	37	43	62	65	39	1	85	14	101	101	117	117
6	63	22	71	38	44	54	20	70	2	86	15	102	102	118	118
7	55	23	46	39	79	55	61	71	3	87	16	103	103	119	119
8	56	24	47	40	60	66	67	72	19	88	17	104	104	120	120
9	64	25	48	41	38	67	69	73	31	89	18	105	105	121	121
10	55	25	73	42	39	68	24	74	30	90	20	106	106	122	122
11	54	27	72	43	61	69	26	75	4	91	21	107	107	123	123
12	68	28	74	44	62	60	27	76	5	92	22	108	108	124	124
13	67	29	43	45	63	61	28	77	6	93	23	109	109	125	125
14	61	30	45	46	32	62	29	78	7	94	24	110	110	126	126
15	62	31	75	47	33	63	30	79	8	95	25	111	111	127	127

【0101】

【表 3 7】

【表 3 7】

【表 5.3】 differential scalefactor arithmetic model 0

寸法 (size)	累積頻度数 (cumulative frequencies)
56	8192, 6144, 5120, 4096, 3072, 2560, 2048, 1792, 1536, 1280, 1024, 896, 768, 640, 576, 512, 448, 384, 320, 288, 256, 224, 192, 176, 160, 144, 128, 112, 96, 88, 80, 72, 64, 56, 48, 44, 40, 36, 32, 28, 24, 22, 20, 18, 16, 14, 13, 12, 11, 10, 9, 8, 7, 6, 0.

【0102】

【表 3 8】

【表 5.4】 differential scalefactor arithmetic model 1

寸法 (size)	累積頻度数 (cumulative frequencies)
67	15018, 13653, 12288, 10922, 10240, 9557, 8874, 8192, 7509, 6826, 6144, 5802, 5461, 5120, 4849, 4778, 4608, 4437, 4266, 4096, 3925, 3840, 3754, 3669, 3584, 3498, 3413, 3328, 3242, 3157, 3073, 2988, 2901, 2816, 2730, 2645, 2560, 2474, 2389, 2304, 2218, 2133, 2048, 1962, 1877, 1792, 1706, 1621, 1536, 1450, 1365, 1280, 1194, 1109, 1024, 938, 853, 768, 682, 597, 512, 426, 341, 256, 170, 85, 0.

【0103】第2番目の方法は、max scalefactor が 8ビットの符号のない整数に符号化される。全てのスケールファクタに対してオフセット値の max scalefactor との差分を算術復号化する。そして、max scalefactor からこの差分信号を引いてスケールファクタを求め

ることができる。差分信号を復号化するために使用される算術モデルは、ビットストリームを構成する1つの要素であり、先にビットストリームで復号化される。

【0104】次のシュドコードは、基本階層と他の高位階層でスケールファクタを復号化する方法を示す。

```

for(ch=0;ch<=nch;ch++)
if(scf_coding[ch]==1)
for(g=0;g<=num_window_group;g++)
for(sfb=layer_sfb_layer;sfb<=layer_sfb_layer+1;sfb++) {
sf_ch_g_sfb=max_scalefactor-arithmetic_decoding();
}
}
else {
for(g=0;g<=num_window_group;g++)
for(sfb=layer_sfb_layer;sfb<=layer_sfb_layer+1;sfb++) {
tmp_index=arithmetic_decoding();
if (tmp_index==54)
tmp_index=54+arithmetic_decoding();
if (sfb==0)
tmp_index=max_scalefactor-tmp_index;
else
tmp_index=sf_ch_g_sfb-1-tmp_index;
sf_ch_g_sfb=index2sf(tmp_index);
}
}
}

```

ここで、layer\_sfb\_layer は、各階層でスケールファクタを復号化するための開始スケールファクタバンドであり、layer\_sfb\_layer+1 が終了スケールファクタバンドである。nch は、チャンネルの数を意味するが、モノラルの場合に1、ステレオの場合に2の値を有する。

【0105】(2, 1, 5, 2) arithmetic model indexの復号化

周波数成分は、無損失符号化を行うために、32個の周

波数係数を含む符号化帯域に分けられる。符号化帯域は、無損失符号化に使用される基本単位である。算術符号化モデル指数は、各符号化帯域のビット分割データを算術符号化／復号化するために使用されるモデルに関する情報である。これは、表4.4に記されたモデルのうちいかなるものを使用しているかを表す。

【0106】

【表39】

【表 4.4】 BSAC Arithmetic Model Parameters

Arithmetic Model index の割当ビット	符号化バンド Listed in 表	Model	Arithmetic Model index の割当ビット	符号化バンド Listed in 表	Model
0	0	表 6.1	16	8	表 6.16
1	—	使用せず	17	8	表 6.17
2	1	表 6.2	18	9	表 6.18
3	1	表 6.3	19	9	表 6.19
4	2	表 6.4	20	10	表 6.20
5	2	表 6.5	21	10	表 6.21
6	3	表 6.6	22	11	表 6.22
7	3	表 6.7	23	11	表 6.23
8	4	表 6.8	24	12	表 6.24
9	4	表 6.9	25	12	表 6.25
10	5	表 6.10	26	13	表 6.26
11	5	表 6.11	27	13	表 6.27
12	6	表 6.12	28	14	表 6.28
13	6	表 6.13	29	14	表 6.29
14	7	表 6.14	30	15	表 6.30
15	7	表 6.15	31	15	表 6.31

【0107】全ての算術符号化モデル指数に対してオフセット値に対する差分が計算され、該差分信号を表4.3に記されたモデルを用いて算術符号化する。このとき、表4.3の4個のモデルのうちのいかなるモデルを使用するかは ArModel model値が表し、ビットストリームに2ビットで保存される。オフセット値は、ビットストリームに min ArModel 値として保存される5ビット

```
for(ch=0;ch<=nch;ch++)
for(sfb=layer sfb layer : sfb<=layer sfb layer+1 : sfb++)
for (g=0 : g<=num window group;g++) {
band= (sfb*num window group)+g
for(i=0;swb offset band : i<=swb offset band+1 : i+=4) {
cband=index2cb (g, i);
if (!decode cband ch g cband ) {
ArModel ch g cband =min ArModel+arithmetic decoding ();
decode cband ch g cband =1 ;
}
}
}
```

ここで、layer sfb layer は、各階層で算術符号化モデル指数を復号化するための開始スケールファクタバンドであり、layer sfb layer+1 は終了スケールファクタバンドである。そして、decode cband ch) g cband は、算術符号化モデルが復号化したか（1）、或いはそうでないか（0）を表すフラグである。

【0109】（2. 1. 6）ビット分割データの復号化

量子化したシーケンスは、ビット分割されたシーケンスから作られる。各4次元ベクトルは、状態に応じて2個

トである。従って、この符号化過程の逆順の差分信号に対する復号化を行い、差分信号とオフセット値とを和して算術符号化モデル指数を復元することができる。

【0108】次のシュドコードは、各階層で算術符号化モデル指数、ArModel cband をいかに復号化するかを示す。

のサブベクトルに更に分けられる。2個のサブベクトルに対する効率の良い圧縮を行うための無損失符号化として算術符号化を行う。各符号化帯域の算術符号化に使用されるモデルを決定し、この情報が算術符号化モデル指数、ArModel に保存される。

【0110】表6.1ないし表6.31に示すように、各算術符号化モデルは多数の下位モデルで構成される。サブベクトルは、算術モデルの可能な多数の下位モデルのうちのいずれか1つを用いて符号化される。下位モデルは、符号化を行うサブベクトルの次元、ベクトルの重

要性、及び各サンプルの符号化状態等によって分類される。ベクトルの重要性は、符号化するベクトルのビット位置によって決定される。即ち、ビット分割されたとき、分割されたビット情報がMSBに対するものか、その次のMSBに対するものか、それともLSBに対するものかによって決まり、MSBが最高の重要性を有し、LSBが最低の重要性を有する。各サンプルの符号化状態は、MSBからLSBへベクトルを符号化しながらその値が更新される。最初はその値が0に初期化される。

[表 6.2] BSAC Arithmetic Model 2

割当ビット=1

snf	pre_state	dimension	累積順位数
1	0	4	14868, 13706, 12646, 11645, 10434, 9479, 8475, 7619, 6467, 5456, 4497, 3601, 2600, 1720, 862, 0

[0 1 1 3]

そして、ビットの値が0でない値になったときに1になる。

[0 1 1 1]

表 6. 1 BSAC Arithmetic Model 0

割当ビット(Allocated bit)=0

BSAC arithmetic model 1

使用せず (not used)

[0 1 1 2]

[表 4 0]

[表 6.3] BSAC Arithmetic Model 3

割当ビット=1

snf	Pre_state	dimension	累積順位数
1	0	4	6476, 4279, 3542, 3269, 2545, 2435, 2199, 2111, 850, 739, 692, 550, 185, 132, 21, 0

[表 4 1]

[0 1 1 4]

[表 6.4] BSAC Arithmetic Model 4

割当ビット=2

snf	pre_state	Dimension	累積順位数
2	0	4	4299, 3445, 2693, 2478, 1669, 1479, 1371, 1333, 450, 347, 245, 219, 81, 50, 15, 0
1	0	4	15290, 14589, 13494, 12485, 11659, 10627, 9583, 8626, 7691, 6727, 5767, 4655, 3646, 2583, 1415, 0
		3	18139, 13484, 11909, 9716, 8068, 6919, 3690, 0
		2	14008, 10384, 8834, 0
		1	11228, 0
	1	4	10365, 9160, 7563, 7004, 6671, 4902, 4183, 3483, 1906, 1661, 1343, 1122, 786, 714, 238, 0
		3	8828, 6616, 4468, 3686, 1769, 1062, 321, 0
		2	4631, 2696, 783, 0
		1	968, 0

[表 4 2]

[0 1 1 5]

[表 4 3]



【表 6.6】BSAC Arithmetic Model 6

割当ビット=2

snf	pre_state	Dimension	累積頻度数
2	0	4	8119, 2306, 1878, 1619, 1076, 1051, 870, 826, 283, 231, 198, 197, 27, 26, 1, 0
1	0	4	3691, 2897, 2406, 2142, 1762, 1668, 1497, 1404, 502, 463, 369, 368, 131, 102, 16, 0
		3	11109, 8893, 6517, 4967, 2735, 2200, 608, 0
		2	10771, 6410, 2819, 0
		1	8112, 0
	1	4	11484, 10106, 7809, 7043, 5063, 3531, 2756, 2603, 2296, 2143, 1990, 1531, 765, 459, 153, 0
		3	10628, 8950, 6018, 4565, 2858, 2129, 786, 0
		2	7595, 4499, 1512, 0
		1	4165, 0

【0116】

【表 4 4】

【表 6.6】BSAC Arithmetic Model 6

割当ビット=3

snf	pre_state	Dimension	累積頻度数
3	0	4	2845, 2371, 1684, 1524, 918, 833, 760, 729, 200, 198, 180, 178, 27, 26, 1, 0
2	0	4	1621, 1183, 983, 776, 646, 628, 516, 484, 210, 207, 188, 186, 39, 36, 1, 0
		3	8800, 6734, 4886, 3603, 1326, 1204, 104, 0
		2	8869, 5163, 1078, 0
		1	3575, 0
	1	4	12803, 12130, 10082, 8767, 8979, 8034, 7404, 6144, 4258, 3780, 3160, 2363, 1575, 945, 630, 0
		3	10410, 8922, 6694, 4270, 2658, 1601, 533, 0
		2	8459, 5107, 1670, 0
		1	4003, 0
1	0	4	5185, 4064, 3423, 3010, 2406, 2289, 2169, 2107, 650, 539, 446, 419, 37, 31, 15, 0
		3	13614, 11020, 8596, 6466, 4845, 3250, 1294, 0
		2	13521, 8754, 4636, 0
		1	9878, 0
	1	4	14091, 12322, 11247, 10389, 8928, 7954, 6696, 6034, 4766, 4023, 3119, 2508, 1594, 1003, 852, 0
		3	12390, 10427, 7006, 6003, 3782, 2580, 926, 0
		2	10008, 6213, 2350, 0
		1	5614, 0

【0117】

【表 4 5】

【表 6.7】BSAC Arithmetic Model 7

割当ビット=3

snf	pre_state	Dimension	累積確率値
3	0	4	3833, 3167, 2542, 2290, 1676, 1606, 1383, 1327, 468, 434, 377, 349, 117, 93, 20, 0
2	0	4	5631, 5620, 4784, 4834, 8363, 3307, 2823, 2882, 1700, 1458, 1213, 1040, 808, 431, 191, 0
		3	11369, 9466, 7519, 6138, 3544, 2441, 1136, 0
		2	11083, 7446, 3439, 0
		1	8823, 0
	1	4	12027, 11579, 9947, 9687, 9239, 8126, 7216, 6176, 4161, 3705, 3055, 3210, 1235, 780, 455, 0
		3	9566, 7943, 4894, 3847, 2263, 1596, 553, 0
		2	7212, 4217, 1240, 0
		1	3296, 0
1	0	4	14363, 13143, 12054, 11153, 10220, 9888, 8609, 7680, 6344, 5405, 4378, 3523, 2762, 1932, 1099, 0
		3	14788, 12256, 11696, 9277, 7581, 6895, 8348, 0
		2	14050, 10293, 6547, 0
		1	10948, 0
	1	4	13856, 12390, 11161, 10156, 8616, 7913, 6939, 6214, 4836, 4062, 3119, 2505, 1624, 1020, 378, 0
		3	12083, 9880, 7293, 6875, 5501, 3372, 628, 0
		2	8773, 5285, 1799, 0
		1	4452, 0

【0118】

【表 4 6】

【表 6.8】 BSAC Arithmetic Mode 18

割当ビット=4

anf	pre_state	Dimension	累積残度数
4	0	4	2770, 3075, 1635, 1511, 1059, 1055, 925, 923, 204, 202, 190, 189, 9, 6, 1, 0
3	0	4	1810, 1254, 1151, 1020, 788, 785, 767, 768, 139, 139, 132, 14, 13, 1, 0
		3	7113, 4896, 3928, 3193, 1096, 957, 97, 0
		2	5858, 4547, 631, 0
		1	4028, 0
	1	4	13263, 10973, 10142, 9752, 8682, 7801, 5851, 5071, 3510, 3120, 2730, 2340, 1560, 780, 390, 0
		3	12976, 11275, 7846, 6356, 4096, 2876, 1097, 0
		2	9479, 5781, 1840, 0
		1	3697, 0
2	0	4	2600, 1768, 1459, 1292, 939, 953, 921, 916, 236, 238, 205, 202, 32, 30, 8, 0
		3	10797, 8340, 6149, 5060, 2371, 1697, 483, 0
		2	10571, 6943, 2445, 0
		1	7864, 0
	1	4	14865, 13983, 11597, 10398, 9385, 8683, 7569, 6899, 5451, 4721, 3484, 3007, 1883, 1208, 690, 0
		3	12611, 10374, 8035, 6167, 4019, 2606, 597, 0
		2	10043, 6306, 2873, 0
		1	6766, 0
1	0	4	6155, 5057, 4328, 3845, 3184, 2977, 2728, 2690, 1841, 1055, 863, 764, 808, 185, 74, 0
		3	12807, 10407, 8143, 6383, 4928, 3013, 1226, 0
		2	18131, 9420, 4028, 0
		1	10396, 0
	1	4	14536, 13348, 11819, 11016, 9340, 8399, 7135, 6671, 5114, 4559, 3821, 3298, 1758, 1177, 438, 0
		3	12735, 10606, 7861, 5011, 3396, 2637, 917, 0
		2	9831, 5972, 2251, 0
		1	4944, 0

【0119】

【表 4 7】

【表 6.9】 BSAC Arithmetic Model 9

割当ビット=4

anf	pre_state	Dimension	累積残度数
4	0	4	3383, 2550, 1967, 1794, 1301, 1249, 1156, 1118, 340, 299, 247, 319, 31, 64, 15, 0
3	0	4	7246, 6275, 5899, 4935, 2771, 3505, 2562, 2814, 1285, 1143, 980, 860, 310, 330, 75, 0
		3	5681, 7809, 6973, 4892, 2774, 1782, 828, 0
		2	11455, 7068, 3368, 0
		1	9437, 0
	1	4	12503, 9701, 8838, 8407, 6698, 5036, 4537, 3664, 2802, 2548, 2371, 2153, 1323, 431, 215, 0
		3	11268, 9422, 6508, 5277, 3078, 2460, 1457, 0
		2	7681, 4565, 1508, 0
		1	2639, 0
2	0	4	11210, 9646, 8423, 7389, 6252, 5746, 5140, 4692, 3860, 3850, 3416, 3014, 1940, 861, 404, 0
		3	12143, 10350, 7784, 6445, 3954, 2528, 1228, 0
		2	10881, 7310, 3874, 0
		1	9337, 0
	1	4	14988, 13408, 11880, 10864, 9681, 8993, 7834, 7196, 6618, 4798, 3671, 2076, 1926, 1312, 827, 0
		3	12488, 10041, 7461, 5732, 3669, 2361, 940, 0
		2	8342, 5847, 1963, 0
		1	5140, 0
1	0	4	14152, 13258, 12486, 11635, 11040, 10290, 9740, 8673, 7646, 6643, 5903, 4928, 4006, 3973, 1761, 0
		3	14895, 13534, 12007, 9787, 8053, 5781, 3570, 0
		2	14086, 10108, 6749, 0
		1	11041, 0
	1	4	14317, 13545, 12244, 11281, 10012, 8962, 7959, 7128, 5791, 4970, 3997, 3128, 2105, 1353, 623, 0
		3	12973, 10678, 8257, 6573, 4186, 2773, 1053, 0
		2	8969, 6039, 2363, 0
		1	5694, 0

【0120】

【表 4 8】

[表 6.10] BSAC Arithmetic Model 10  
 例当ビット (Abit) = 5

snf	pre_state	Dimension	累積値
Abit	0	4	2326, 1619, 1271, 1777, 501, 892, 841, 832, 141, 146, 120, 129, 24, 23, 1, 0
Abit	0	4	1746, 1251, 1038, 968, 615, 611, 583, 582, 106, 104, 101, 99, 3, 2, 1, 0
-1		3	7110, 6230, 4228, 3552, 658, 632, 46, 0
		2	8101, 2575, 265, 0
		1	1489, 0
	1	4	13010, 12047, 11563, 11083, 9637, 8673, 6754, 5782, 4336, 3855, 3373, 2891, 2409, 1927, 963, 0
		3	10635, 10132, 8316, 7158, 5595, 3428, 2318, 0
		2	8208, 5197, 1287, 0
		1	4954, 0
Abit	0	4	2137, 1660, 1471, 1312, 1007, 1000, 957, 951, 803, 278, 249, 247, 48, 47, 1, 0
-2		3	9327, 7413, 5073, 4291, 2027, 1695, 206, 0
		2	9658, 5404, 1628, 0
		1	5660, 0
	1	4	13360, 12288, 10727, 9752, 8484, 7899, 7119, 6521, 5253, 3900, 3023, 2536, 1852, 1257, 685, 0
		3	13742, 11685, 8877, 7230, 5015, 3426, 1132, 0
		2	10403, 6691, 2828, 0
		1	5298, 0
Abit	0	4	4124, 3181, 2702, 2519, 1859, 1922, 1733, 1712, 524, 475, 425, 407, 78, 52, 15, 0
-3		3	10829, 8581, 6286, 4863, 2539, 1920, 594, 0
		2	11074, 7282, 3092, 0
		1	8045, 0
	1	4	14641, 13343, 11637, 10662, 9325, 8783, 7213, 6517, 5485, 5033, 4118, 3505, 2143, 1656, 809, 0
		3	13010, 11143, 8682, 7202, 4537, 3297, 1221, 0
		2	9941, 5861, 2191, 0
		1	5340, 0
Other	0	4	9846, 8235, 7136, 6401, 5551, 5131, 4664, 4320, 3908, 3399, 1878, 1508, 935, 603, 277, 0
rsnf		3	13070, 11424, 9084, 7203, 4771, 3479, 1486, 0
		2	18169, 9298, 6406, 0
		1	10371, 0
	1	4	14766, 13666, 12368, 11442, 10035, 9078, 7967, 7045, 5824, 5008, 4058, 3400, 2353, 1612, 659, 0
		3	13391, 11189, 8904, 7172, 4956, 3183, 1363, 0
		2	10280, 6372, 2633, 0
		1	5419, 0

[0121]

[表 49]

【表 6.11】 BSAC Arithmetic Model 11

割当ビット (Abit) =6

snf	pre_state	Dimension	乗数値
Abit -1	0	4	2872, 2294, 1740, 1592, 1341, 1155, 1035, 960, 819, 800, 251, 247, 105, 77, 34, 0
		4	3854, 3090, 2489, 2278, 1801, 1655, 1548, 1505, 827, 829, 446, 400, 193, 141, 61, 0
		3	10554, 8555, 6878, 4978, 3556, 2219, 523, 0
		2	10569, 6180, 3696, 0
1	1	1	8971, 0
		4	11419, 11170, 10922, 10438, 7942, 6650, 8722, 3478, 1737, 1489, 1341, 952, 744, 496, 248, 0
		3	11013, 8245, 6730, 4982, 3283, 1689, 553, 0
		2	6969, 4370, 1866, 0
Abit -2	0	1	2156, 0
		4	2603, 8070, 6343, 6474, 5305, 5009, 4290, 4029, 3333, 1911, 1581, 1893, 668, 448, 217, 0
		3	11539, 9620, 7628, 6280, 4013, 2653, 1021, 0
		2	12453, 8284, 4723, 0
1	1	1	9182, 0
		4	18472, 12596, 10499, 9187, 7990, 7464, 6063, 9008, 4614, 3747, 2518, 2477, 1641, 1084, 557, 0
		3	13059, 10826, 8476, 6915, 4488, 2956, 1323, 0
		2	9212, 5772, 3053, 0
Abit -3	0	1	4244, 0
		4	14183, 12785, 11663, 10840, 9601, 8754, 8135, 7363, 6014, 5237, 4482, 3727, 2708, 1818, 888, 0
		3	13654, 11914, 9714, 7830, 5717, 3916, 2112, 0
		2	12497, 8801, 4969, 0
1	1	1	10296, 0
		4	14068, 18770, 18294, 11913, 10230, 8388, 8439, 7428, 6296, 5368, 4381, 3620, 2594, 1797, 895, 0
		3	13120, 10879, 8448, 6665, 4366, 2794, 1047, 0
		2	9311, 5378, 1792, 0
Othe r snf	0	1	4655, 0
		4	15173, 14794, 14389, 13659, 13224, 12600, 11964, 11067, 10197, 9375, 9081, 7824, 6697, 4691, 3216, 0
		3	15229, 12655, 12748, 10084, 8587, 6459, 4111, 0
		2	14661, 11170, 7924, 0
1	1	1	11389, 0
		4	14873, 13768, 12488, 11491, 10239, 9164, 7990, 7189, 5992, 5012, 4119, 3389, 2228, 1457, 524, 0
		3	13083, 10913, 8477, 6753, 4529, 3047, 1244, 0
		2	10101, 6349, 2615, 0
1	1	1	6350, 0

【0122】表 6. 12 BSAC arithmetic model 12  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=6

【0123】表 6. 13 BSAC arithmetic model 13  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=6

【0124】表 6. 14 BSAC arithmetic model 14  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=7

【0125】表 6. 15 BSAC arithmetic model 15  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=7

【0126】表 6. 16 BSAC arithmetic model 16  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=8

【0127】表 6. 17 BSAC arithmetic model 17  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=8

【0128】表 6. 18 BSAC arithmetic model 18  
BSAC arithmetic model 10と同一、但し、割当ビット(A

bit)=9

【0129】表 6. 19 BSAC arithmetic model 19  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=9

【0130】表 6. 20 BSAC arithmetic model 20  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=10

【0131】表 6. 21 BSAC arithmetic model 21  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=10

【0132】表 6. 22 BSAC arithmetic model 22  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=11

【0133】表 6. 23 BSAC arithmetic model 23  
BSAC arithmetic model 11と同一、但し、割当ビット(A  
bit)=11

【0134】表 6. 24 BSAC arithmetic model 24  
BSAC arithmetic model 10と同一、但し、割当ビット(A  
bit)=12

【0135】表 6. 25 BSAC arithmetic model 25

BSAC arithmetic model 11と同一、但し、割当ビット(A bit)=12

【0136】表6.26 BSAC arithmetic model 26  
BSAC arithmetic model 10と同一、但し、割当ビット(A bit)=13

【0137】表6.27 BSAC arithmetic model 27  
BSAC arithmetic model 11と同一、但し、割当ビット(A bit)=13

【0138】表6.28 BSAC arithmetic model 28  
BSAC arithmetic model 10と同一、但し、割当ビット(A bit)=14

【0139】表6.29 BSAC arithmetic model 29  
BSAC arithmetic model 11と同一、但し、割当ビット(A bit)=14

【0140】表6.30 BSAC arithmetic model 30  
BSAC arithmetic model 10と同一、但し、割当ビット(A

```
pre_state = State that indicates whether the current decoded value is 0 or not.
snf = the significance of the vector to be decoded.
idx0 = codeword index whose previous states are 0
idx1 = codeword index whose previous states are 1
dec_sample = data to be decoded
start_i = start frequency line of the decoded vectors.
for(i=start_i; i<start_i+4; i++) {
    if(pre_state) {
        if(idx1 & 0x01)
            dec_sample[i] = (1<<(snf-1))
            idx1<>1;
    }
    else {
        if(idx0 & 0x01)
            dec_sample[i] = (1<<(snf-1))
            idx0<>1;
    }
}
```

【0143】量子化した周波数係数のビット分割されたデータがMSBからLSBへ符号化されつつ、最初に0でない値が符号化される時、0でない周波数係数に対する符号ビットは算術符号化する。符号ビットは真数のときは1を表し、正数のときは0を表す。従って、復号化器においても、ビット分割データに対する算術復号化が行われ、算術復号化されたビット値が最初に0でない場合、ビットストリームで符号に関する情報のacode\_signが相次ぐ。この情報を以てsign\_bitを、表5.13に示すモデルを以て算術復号化することができる。そこで、sign\_bitが1であれば、先に分離されたデータを結合して作られた量子化したデータ(y)に符号に関する情報を、次のように与えることができる。

```
if(y!=0)
if(sign_bit==1)
```

bit)=15

【0141】表6.31 BSAC arithmetic model 31  
BSAC arithmetic model 11と同一、但し、割当ビット(A bit)=15

【0142】2つに分けられたサブベクトルは、1ないし4次元間のベクトルになる。このサブベクトルは、MSBからLSBへ低い周波数から高い周波数成分へ進みつつ算術符号化される。算術符号化に使用される算術符号化モデル指数は、符号化帯域単位でビット分割データを伝送するに先立ち、低い周波数から高い周波数順に予めビットストリームへ保存される。各ビット分割されたデータに対する算術復号化の結果は、コードワードインデキストである。このインデキストは、以下のようなシュドコードによるビット結合によって元の量子化したデータに復元される。

y = -y

【0144】(2.2) M/Sステレオ処理部(選択モジュール)

ビットストリームに含まれているフラグのms\_usedに基づき、各スケールファクタバンドのM/Sステレオ処理部150の使用可否が分かり、使用された場合、AACに記されている過程に従って処理する。

(2.3) 予測部(選択モジュール)

ビットストリームに含まれているフラグのprediction\_presentに基づき、予測モジュールの使用可否が分かる。使用されたならば、AACに記されている過程に従って処理する。

(2.4) インテンシティステレオ処理部(選択モジュール)

ビットストリームに含まれているフラグのstereo\_info

に基づき、各スケールファクタバンドのインテンシティステレオ処理部130の使用可否が分かる。使用された場合、AACに記されている過程に従って処理する。

(2.6) TNS部(選択モジュール)

ビットストリームに含まれているフラグのtns\_presentに基づき、TNS部の使用可否が分かる。使用された場合、AACに記されている過程に従って処理する。

(2.6) 逆量子化部

逆量子化部320では、復号化したスケールファクタと量子化したデータとを以て元の寸法の信号に復元される。逆量子化過程は、AAC標準案に逆量子化の過程が記されている。

(2.7) 周波数/時間マッピング

周波数/時間マッピング部370は、周波数領域のオーディオ信号を改めて時間領域の信号に切替え、使用者が再生できるようにする。周波数信号を時間領域の信号にマッピングする式は、AAC標準案に定義されている。又、マッピングに係わったウィンドウ等、各種の関連事項がAAC標準案に共に記されている。

【0145】一方、本実施形態は、コンピュータで実行できるプログラムによって作成可能である。そして、コンピュータで使用される媒体からプログラムを動作させる汎用のデジタルコンピュータで具現できる。前記媒体は、マグネチック貯蔵媒体(例えば、ROM、フロッピーディスク、ハードディスク等)、光学的判読媒体(例えば、CD-ROM、DVD等)及びキャリアウェーブ(例えば、インターネットを介して伝送)のような貯蔵媒体を含む。

【0146】例えば、前記記録媒体は、オーディオ信号を基本階層と予め決定された数の高位階層とからなる階層的なビットストリームに符号化するビット率の調節可能なオーディオ信号符号化方法において、入力オーディオ信号を信号処理し、所定の量子化帯域別に量子化する量子化処理を行うプログラムコード手段と、前記量子化処理されたデータのうちの基本階層に相応する量子化したデータを符号化する基本階層符号化処理をコンピュータによって実行しうるプログラムコード手段と、前記符号化した階層の次の高位階層に属する量子化したデータ及び前記符号化した階層に属しつつ、階層の大きさの制限を理由に符号化できずに残っている量子化したデータを符号化する階層符号化処理をコンピュータによって実行しうるプログラムコード手段、及び前記階層符号化処理を全階層に対して行い、ビットストリームに形成するビットストリーム形成処理をコンピュータによって実行しうるプログラムコード手段とを含む。前記基本階層符号化処理、階層符号化処理及びビットストリーム形成処理の符号化は、符号化しようとする階層に相応する付加情報及び量子化したデータを所定個数のビットで表現し、重要度が最高のビットからなる最上位ビットシーケンスから重要度の低いビットからなるビットシーケンス

の順に従って所定の確率モデルを用いて算術符号化するものの、但し、ビット分割された左チャンネルデータと右チャンネルデータとを所定のベクトル単位に交互に符号化する。そして、本発明を具現するための機能的プログラム、コード及びコードセグメントは、本発明が属する技術分野のプログラマによって容易に推論できる。

【0147】

【発明の効果】以上詳細に説明したように、本発明によれば、次の(a)～(j)のような効果がある。

(a) 種々の使用者の要求に応えるため、モノラル信号やステレオ信号を一率に処理できるように、ビットストリームの構成を柔軟ならしめつつも、高いビット率での性能は既存の圧縮のみ考慮した場合の符号化器と類似の性能を提供する。即ち、使用者の要求に応じて多数の階層のビット率に関する情報を重複無しに1本のビットストリームに組合わせることにより、特に高いビット率で良好な音質のビットストリームを提供できる。

(b) 送信端同士の間接器を必要とせず、伝送経路の状態や使用者のいかなる要求も受入れられるという効果がある。

(c) モノラル信号にのみ適用されたスケラビリティの機能をステレオ信号にまで広げることにより、スケラビリティの適用範囲を一層広げることができる。

(d) 既存の符号化/復号化能を改良するためのモジュールを含むオーディオ符号化/復号化器にも適用できることから、各ビット率での性能の改良を成し遂げることができる。

(e) MPEG-2 AAC標準案等で使用される時間/周波数マッピングや量子化等の基本的なモジュールのみならず、性能を改良するための種々のモジュールを受入れつつ、無損失符号化する部分のみを別にして使用することにより、スケラビリティを提供できる。

(f) ビット率が調節可能な形のビットストリームであることから、1本のビットストリーム内に多数のビット率のビットストリームを含んでいる。他のスケラブル符号化器とは異なり、微細な間隔階層を提供することができ、既存のスケラブルコーデックに比べて活用範囲をはるかに広げることができる。

(g) 従来のスケラブルオーディオコーデックとは異なり、高いビット率で良好な音質を提供する。AAC標準案と組合わせられるならば、最高位階層のビット率ではAACとほぼ同一の音質にできる。

(h) AACと殆ど同一で、単に無損失符号化部分のみが異なるため、AACビットストリームで周波数領域の量子化した信号を復号化し、この復号化した信号を基に、改めてBSACスケラブルビットストリームに作ることができる。即ち、無損失変号化が可能である。また逆に、BSACスケラブルビットストリームからAACビットストリームを作ることができる。このような機能性のため、符号化効率のみのために作られた多くのAACビ

ットストリームが状況に合うように切替えられて使用される。従って、スケーラビリティを提供するために別途のスケーラブル符号化器でスケーラビリティ提供用ビットストリームを提供する等、無駄な手間を回避できる。

(i) 従来の符号化方式のように定めてあるビット率で最上の性能を示す符号化効率が良好な技法であるばかりか、マルチメディア時代に即して符号化したビット率内のビット率の1つに復元できる符号化/復号化技法である。

(j) 多数の階層のビット率に関するデータを1本のビットストリーム内に表現することにより、伝送経路の都合、復号化器の性能又は使用者の要求事項に応じてビ

ットストリームの大きさを変えることができ、復号化器の複雑性も変えることができる。

【図面の簡単な説明】

【図1】 本発明に係る符号化器の構造を示す図である。

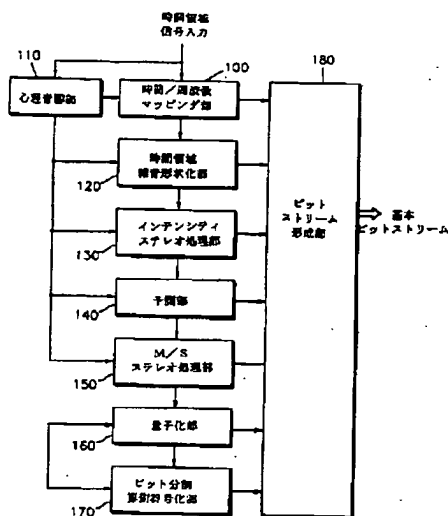
【図2】 本発明に係るビットストリームの構造を示す図である。

【図3】 本発明に係る復号化器の構造を示す図である。

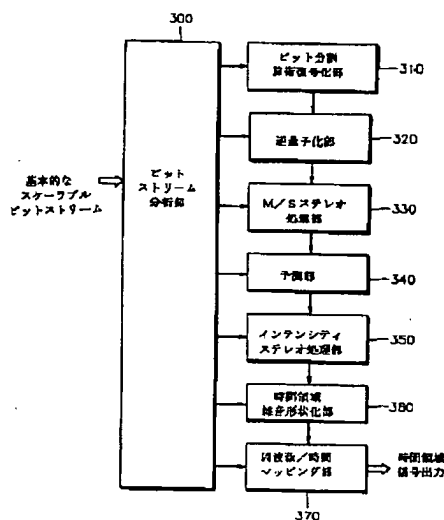
【図4】 長いブロック（ウィンドウ法=2048）において周波数成分配列を示す図である。

【図5】 短いブロック（ウィンドウ法=256）において周波数成分配列を示す図である。

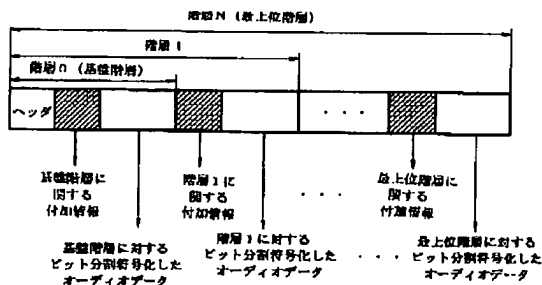
【図1】



【図3】



【図2】





【図 4】



【図 5】

